

Ashish Ghosh
Rajarshi Pal
Rajendra Prasath (Eds.)

LNAI 10682

Mining Intelligence and Knowledge Exploration

5th International Conference, MIKE 2017
Hyderabad, India, December 13–15, 2017
Proceedings

 Springer

Lecture Notes in Artificial Intelligence

10682

Subseries of Lecture Notes in Computer Science

LNAI Series Editors

Randy Goebel

University of Alberta, Edmonton, Canada

Yuzuru Tanaka

Hokkaido University, Sapporo, Japan

Wolfgang Wahlster

DFKI and Saarland University, Saarbrücken, Germany

LNAI Founding Series Editor

Joerg Siekmann

DFKI and Saarland University, Saarbrücken, Germany

More information about this series at <http://www.springer.com/series/1244>


Ashish Ghosh · Rajarshi Pal
Rajendra Prasath (Eds.)

Mining Intelligence and Knowledge Exploration

5th International Conference, MIKE 2017
Hyderabad, India, December 13–15, 2017
Proceedings

Editors

Ashish Ghosh
Indian Statistical Institute
Kolkata
India

Rajendra Prasath 
Indian Institute of Information Technology
Sri City
India

Rajarshi Pal
Institute for Development and Research in
Banking Technology
Hyderabad
India

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Artificial Intelligence
ISBN 978-3-319-71927-6 ISBN 978-3-319-71928-3 (eBook)
<https://doi.org/10.1007/978-3-319-71928-3>

Library of Congress Control Number: 2017960852

LNCS Sublibrary: SL7 – Artificial Intelligence

© Springer International Publishing AG 2017

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume contains the papers presented at MIKE 2017: the 5th International Conference on Mining Intelligence and Knowledge Exploration held during December 13–15, 2017, at the Institute for Development and Research in Banking Technology (IDRBT), Hyderabad, India (<http://www.mike.org.in/2017/>). MIKE 2017 received 139 qualified submissions from 17 countries and each qualified submission was reviewed by a minimum of three Program Committee members using the criteria of relevance, originality, technical quality, and presentation. A rigorous review process with the help of an illustrious Program Committee led to 40 of these submissions being accepted for presentation at the conference. Hence, the overall acceptance rate for this edition of MIKE is 28.78%.

The International Conference on Mining Intelligence and Knowledge Exploration (MIKE) is an initiative focusing on research and applications on various topics of human intelligence mining and knowledge discovery. Human intelligence has evolved steadily over several generations, and today human expertise is excelling in multiple domains and in knowledge-acquiring artifacts. The primary goal was to focus on the frontiers of human intelligence mining toward building a body of knowledge in this key domain. The focus was also to present state-of-art scientific results, to disseminate modern technologies, and to promote collaborative research in mining intelligence and knowledge exploration. At MIKE 2017, specific emphasis was placed on the “learning to explore smart and intelligent systems.”

MIKE 2017 identified nine tracks topic wise, each led by two to three track coordinators (in total, there were 23 track coordinators) to contribute and also to handle submissions falling in their areas of interest. The enthusiastic involvement from each of them along with the supervision of the program chairs ensured selection of only quality papers for the conference. Each track coordinator took enormous responsibility to fulfil the tasks assigned to them since we started circulating the first call for papers. This is reflected in every paper in the proceedings and had a huge impact on the quality of the submissions.

The accepted papers were chosen on the basis of research excellence, which provides a body of literature for researchers involved in exploring, developing, and validating learning algorithms and knowledge-discovery techniques. Accepted papers were grouped into various subtopics including artificial intelligence, machine learning, image processing, pattern recognition, speech processing, information retrieval, natural language processing, social network analysis, security, fuzzy rough sets, and other areas. Researchers presented their work and had an excellent opportunity to interact with eminent professors and scholars in their area of research. All participants benefited from discussions that facilitated the emergence of new ideas and approaches.

We were pleased to have the following dignitaries serving as advisory members for MIKE 2017: Prof. Ramon Lopaz de Mantaras, Artificial Intelligence Research Institute, Spain; Prof. Mandar Mitra, Indian Statistical Institute (ISI), Kolkata, India;

Prof. Agnar Aamodt, Pinar Ozturk and Prof. Bjorn Gambäck, Norwegian University of Science and Technology, Norway; Prof. Sudeshna Sarkar and Prof. Niloy Ganguly, Indian Institute of Technology, Kharagpur, India, Prof. Philip O'Reilly, University College Cork, Ireland; Prof. Nirmalie Wiratunga, Robert Gordon University, UK; Prof. Paolo Rosso, Universitat Politècnica de Valencia, Spain; Prof. Chaman L. Sabharwal, Missouri University of Science and Technology, USA; Prof. Tapio Saramaki, Tampere University of Technology, Finland; Prof. Vasudeva Verma, IIIT Hyderabad, India; Prof. Niloy Ganguly, Indian Institute of Technology, Kharagpur, India; Prof. Grigori Sidorov, NLTP Laboratory CIC - IPN, Mexico; Prof. Genoveva Vargas-Solar, CNRS, France; Prof. Ildar Batyrshin, National Polytechnic Institute, Mexico; Dr. Kazi Shah Nawaz Ripon, NTNU, Trondheim, Norway; and Dr. Krishnaiyya Jallu, Bharat Heavy Electronics Limited, Thiruchirappalli, India.

We sincerely express our gratitude to Prof. B. Yegnanarayana, INSA Senior Scientist, International Institute of Information Technology, Hyderabad, and Prof. Chaman Lal Sabharwal, Missouri University of Science and Technology, Rolla, USA, for being the general chairs. Their guidance, suggestions, and constant support were invaluable in planning the various activities of MIKE 2017.

Several eminent scholars — including Prof. Sankar Kumar Pal, Distinguished Scientist and Former Director, Indian Statistical Institute, Kolkata; Prof. Sung-Bae Cho, Yonsei University, Korea; Prof. Alexander Gelbukh, Instituto Politécnico Nacional, Mexico; and Prof. N. Subba Reddy, Gyeongsang National University, Jinju, Korea — delivered invited talks on various topics of artificial intelligence, machine learning, and soft computing.

We also organized two workshops on (a) Artificial Intelligence for Banking and Finance, organized by Dr. Rajarshi Pal of IDRBT, Hyderabad, and (b) Deep Learning and Industrial Applications, organized by Dr. Krishnaiah Jallu of BHEL, Hyderabad. These two workshops helped to motivate the young and aspiring researchers to participate in active research work.

A large number of eminent professors, well-known scholars, industry leaders, and young researchers participated in making MIKE 2017 a great success. We recognize and appreciate the hard work of each author of the articles published in these proceedings. We also express our sincere thanks to the Institute for Development and Research in Banking Technology (IDRBT), Hyderabad, for allowing us to host MIKE 2017.

We thank the Technical Program Committee members and all reviewers for their timely and thorough participation in the reviewing process. We express our sincere gratitude to Shri Harun R. Khan, Former Deputy Governor, Reserve Bank of India, who kindly agreed to be the chief guest for the inauguration ceremony of MIKE 2017 as well as Dr. A. S. Ramasastri, Director, IDRBT, Hyderabad, for his encouragement and support in organizing MIKE 2017 in IDRBT, Hyderabad this year. We appreciate the time and effort invested by the members of the local organizing team at IDRBT, Hyderabad, and IIIT Sricity. We are very grateful to all our sponsors for their generous support of MIKE 2017.

Finally, we acknowledge the use of EasyChair in the submission, review, and proceedings creation processes.

We are very pleased to express our sincere thanks to Springer staff, especially Alfred Hofmann, Anna Kramer, and the editorial team, for their faith and support in publishing the proceedings of MIKE 2017.

December 2017

Ashish Ghosh
Rajarshi Pal
Rajendra Prasath

Organization

Program Committee

Amit A Nanavati	IBM Research, India
Agnar Aamodt	Norwegian University of Science and Technology, Norway
Arif Ahmed	Haldia Institute of Technology, West Bengal, India
Kazi Masudul Alam	University of Ottawa, Canada
Lasker Ershad Ali	Peking University, Beijing, China
Gloria Inés Alvarez	Pontificia Universidad Javeriana Cali, Colombia
Rema Ananthanarayanan	IBM Research, India
M. Gethsiyal Augasta	Kamaraj College, Tuticorin, India
Zeyar Aung	Masdar Institute of Science and Technology, Abu Dhabi, United Arab Emirates
R. Venkatesh Babu	Indian Institute of Science, Bangalore, India
Lavanya Balaraja	University of Madras, Chennai, India
Vineeth Balasubramanian	Indian Institute of Technology, Hyderabad
Biplab Banerjee	Indian Institute of Technology, Roorkee, India
Dip Sankar Banerjee	Indian Institute of Information Technology, Guwahati, India
Anupam Basu	Indian Institute of Technology, Kharagpur, India
Indranil Basu	Institute of Engineering and Management, Kolkata
Tanmay Basu	University of Michigan, Ann Arbor, USA
Laxmidhar Behera	Indian Institute of Technology, Kanpur, India
Vasudha Bhatnagar	University of Delhi, New Delhi, India
Chiranjib Bhattacharya	Indian Institute of Science, Bangalore, India
Dhruba Bhattacharya	Tezpur University, India
Sourangshu Bhattacharya	Indian Institute of Technology, Kharagpur, India
Malay Bhattacharyya	Indian Statistical Institute, Kolkata, India
Pushpak Bhattacharyya	Indian Institute of Technology, Bombay, India
S. Nagesh Bhattu	Institute for Development and Research in Banking Technology (IDRBT), Hyderabad
Plaban Kumar Bhowmik	Indian Institute of Technology, Kharagpur, India
Arindam Biswas	Indian Institute of Engineering Science and Technology, Shibpur
Saroj K. Biswas	National Institute of Technology, Silchar, India
Indranil Bose	Indian Institute of Management, Kolkata, India
Samarjit Bose	Indian Statistical Institute, Kolkata, India
Darko Brodić	University of Belgrade, Serbia
Erik Cambria	Nanyang Technological University, Singapore
Basabi Chakraborty	Iwate Prefectural University, Iwate, Japan

Tanmoy Chakraborty	Indraprastha Institute of Information Technology, New Delhi, India
Snehashish Chakraverty	National Institute of Technology, Rourkela, India
Krishna Mohan Chalavadi	Indian Institute of Technology Hyderabad, India
Jonathan Chan	King Mongkut's University of Technology, Thailand
Bhabatosh Chanda	Indian Statistical Institute, Kolkata, India
Joydeep Chandra	Indian Institute of Technology, Patna, India
Sanjay Chatterji	Indian Institute of Information Technology, Kalyani, India
Samiran Chattopadhyay	Jadavpur University, Kolkata, India
Subhasis Chaudhury	Indian Institute of Technology, Bombay, India
Manoj Kumar Chinnakotla	Microsoft (Bing), Hyderabad, India
Sung-Bae Cho	Yonsei University, Seoul, Korea
Kamal Kumar Choudhary	Indian Institute of Technology, Ropar, India
Ananda S. Chowdhury	Jadavpur University, Kolkata, India
Isis Bonet Cruz	Universidad EIA, Antioquia, Colombia
Guru D. S.	University of Mysore, India
Dipankar Das	Jadavpur University, Kolkata, India
Saurabh Das	Indian Statistical Institute, Kolkata, India
Sudeb Das	Videonetics Pvt Ltd, Kolkata, India
Swagatam Das	Indian Statistical Institute, Kolkata, India
Tirthankar Dasgupta	Tata Consultancy Services, New Delhi, India
Ajaya Kumar Dash	International Institute of Information Technology, Bhubaneswar, India
Aloke Datta	National Institute of Technology, Meghalaya, India
Rajat K. De	Indian Statistical Institute, Kolkata, India
Rameswar Debnath	Khulna University, Bangladesh
Satchidananda Dehuri	Fakir Mohan University, Orissa, India
Maunendra Sankar Desarkar	Indian Institute of Technology, Hyderabad, India
Somnath Dey	Indian Institute of Technology Indore
Abhinav Dhall	Indian Institute of Technology, Ropar
Debi Prosad Dogra	Indian Institute of Technology, Bhubaneswar, India
Irina Dragoste	TU Dresden, Germany
Aidan Duane	Waterford Institute of Technology (WIT), Ireland
Asif Ekbal	Indian Institute of Technology, Patna, India
Debasis Ganguly	IBM Research Labs, Dublin, Ireland
Niloy Ganguly	Indian Institute of Technology, Kharagpur, India
Vinay Gautam	Computer and Information Science, NTNU, Norway
Alexander Gelbukh	Instituto Politécnico Nacional
Ashish Ghosh	Indian Statistical Institute, Kolkata, India
Saptarshi Ghosh	Indian Institute of Technology Kharagpur, India
Sujata Ghosh	Indian Statistical Institute, Chennai, India
Susmita Ghosh	Jadavpur University, India
Rob Gleasure	University College Cork, Ireland
Sumit Goswami	Defence Research and Development Organization, New Delhi, India

Pawan Goyal	Indian Institute of Technology, Kharagpur, India
Adrian Groza	Technical University of Cluj-Napoca, Romania
Phalguni Gupta	Indian Institute of Technology, Kanpur, India
Rajeev Gupta	IBM Research, India
Anindya Halder	North-Eastern Hill University, Shillong, Meghalaya, India
Wu Huayu	Institute for Infocomm Research, Singapore
Prasanta K. Jana	Indian Institute of Technology (ISM) Dhanbad, India
Saroj K. Meher	Indian Statistical Institute, Bangalore, India
Saurav Karmakar	Georgia State University, USA
Byung-Gyu Kim	Sookmyung Women's University, South Korea
Sangwoo Kim	Severance Biomedical Science Institute, Yonsei University College of Medicine, South Korea
P. V. V. Kishore	K. L. University, Guntur, Andhra Pradesh, India
Palanivel Kodeswaran	IBM Research, India
Shruti Kohli	Birla Institute of Technology, Mesra, India
Nagesh Kolagani	Indian Institute of Information Technology, Sri City, India
Nagesh Kumar	Indian Institute of Science, Bangalore, India
T. V. Vijay Kumar	Jawaharlal Nehru University, New Delhi, India
Durairaju Kumaran Raju	Geoscience Consulting Pte Ltd, Singapore
Krishna Kummamuru	IBM Research, India
Ashish Kundu	IBM T.J. Watson Research Center, USA
Malay Kumar Kundu	Indian Statistical Institute, Kolkata, India
Venkatareshbabu Kuppili	National Institute of Technology, Goa, India
Arnab Kumar Laha	Indian Institute of Management Calcutta, Kolkata, India
Uttama Lahiri	Indian Institute of Technology Gandhinagar, India
Chaman Lal Sabharwal	Missouri University of Science and Technology, USA
Helge Langseth	Norwegian University of Science and Technology, Norway
Camelia Lemnaru	Technical University of Cluj-Napoca, Romania
Ramon Lopez De Mantaras	Artificial Intelligence Research Laboratory, IIIA - CSIC, Barcelona, Spain
Yutaka Maeda	Kansai University, Japan
Rajib Ranjan Maiti	Singapore University of Technology and Design, Singapore
Santi Maity	Indian Institute of Engineering Science and Technology, Shibpur, India
Pradipta Maji	Indian Statistical Institute, Kolkata, India
Suman Kumar Maji	Indian Institute of Technology Patna, India
Kaushik Majumdar	Indian Statistical Institute, Bangalore, India
Prasenjit Majumder	DAIICT, Gandhinagar, India
Aradhna Malik	Indian Institute of Technology, Kharagpur, India
Rajib Mall	Indian Institute of Technology, Kharagpur, India

Radhika Mamidi	International Institute of Information Technology, Hyderabad, India
Pikakshi Manchanda	Università degli Studi di Milano-Bicocca, Italy
Anca Marginean	Technical University of Cluj-Napoca, Romania
Abhijit Mishra	IBM Research, India
Mandar Mitra	Indian Statistical Institute, Kolkata
Pabitra Mitra	Indian Institute of Technology, Kharagpur, India
Pralay Mitra	Indian Institute of Technology, Kharagpur
Suman Mitra	DAIICT, Gandhinagar, India
Delia Mitrea	Technical University of Cluj-Napoca, Romania
Vinay Kumar Mittal	Ritwik Software Technologies Pvt. Ltd., Hyderabad
Natwar Modani	Adobe Systems Inc., San Jose, USA
Hans Moen	Norwegian University of Science and Technology, Norway
Ajoy Mondal	Indian Statistical Institute, Kolkata, India
Sougata Mukherjea	IBM Research, India
Snehasis Mukherjee	Indian Institute of Information Technology Chittoor, Sricity, India
C. A. Murthy	Indian Statistical Institute, Kolkata, India
K. Ramachandra Murthy	Institute for Development and Research in Banking Technology (IDRBT), Hyderabad, India
Narasimha Murty	Indian Institute of Science, Bangalore, India
M. Muthuramakrishnan	Singapore University of Technology and Design, Singapore
Madalina Mandy Nagy	Technical University of Munich, Germany
Bilegsaikhon Naidan	Norwegian University of Science and Technology
Tomoharu Nakashima	Osaka Prefecture University
Pradipta Kumar Nanda	Siksha O Anusandhan University, Bhubaneswar, India
Ramasuri Narayanam	IBM Research, India
Mita Nasipuri	Jadavpur University, Kolkata, India
Bhabesh Nath	Tezpur University, India
Atul Negi	University of Hyderabad, India
Naveen Nekuri	University of Hyderabad, India
Jian-Yun Nie	Université de Montréal, Canada
Aditya Nigam	Indian Institute of Technology, Mandi, India
Maciej Ogrodniczuk	Institute of Computer Science, Polish Academy of Sciences, Poland
Sylvester Olubolu Orimaye	Monash University (Australia), Australia
Inah Omoronyia	University of Glasgow, UK
Santiago Ontanon	Drexel University, USA
Pinar Ozturk	Norwegian University of Science and Technology, Norway
Jiaul Paik	Indian Institute of Technology, Kharagpur, India
Partha Pakray	National Institute of Technology Mizoram, India
Rajarshi Pal	Institute for Development and Research in Banking Technology, Hyderabad, India

Sukomal Pal	Indian Institute of Technology (BHU), Varanasi, India
Umapada Pal	Indian Statistical Institute, Kolkata, India
V Pallavi	Philips Research India, Bangalore, India
Marco Palomino	University of Plymouth, UK
Bhawani Panda	Indian Institute of Technology, New Delhi, India
Chhabi Rani Panigrahi	Central University of Rajasthan, India
Ranjani Parthasarathi	Anna University, Chennai, India
Praveen Paruchuri	International Institute of Information Technology, Hyderabad, India
Bibudhendu Pati	Indian Institute of Technology Kharagpur, India
Dipti Patra	National Institute of Technology, Rourkela, India
Soma Paul	International Institute of Information Technology, Hyderabad, India
Maciej Piasecki	Wroclaw University of Technology, Poland
Carla Pires	Universidade Federal de Pelotas (UFPEL), Pelotas, Brazil
Saithi Podila	Georgia State University, USA
Shiraj Pokharel	Georgia State University, USA
Octavian Pop	Technical University of Cluj-Napoca, Romania
M. V. N. K. Prasad	Institute for Development and Research in Banking Technology, Hyderabad
Rajendra Prasath	Indian Institute of Information Technology Chittoor, Sricity, India
Dilip Pratihari	Indian Institute of Technology, Kharagpur, India
Pulak Purkait	Indian Statistical Institute, Kolkata, India
P. V. Rajkumar	Texas Southern University, USA
K. Srinivasa Raju	BITS Pilani Hyderabad Campus, India
S. Bapi Raju	University of Hyderabad, India
Vijay Sundar Ram	AU-KBC Research Centre, Anna University, Chennai, India
K. Ramakrishnan	Pondicherry Engineering College, Pondicherry, India
C. Raggavendra Rao	University of Hyderabad, India
K. Sreenivasa Rao	Indian Institute of Technology, Kharagpur, India
V. Ravi	Institute for Development and Research in Banking Technology (IDRBT), Hyderabad
Shubhra Sankar Ray	Indian Statistical Institute, Kolkata, India
Juan Recio-Garcia	Universidad Complutense de Madrid, Spain
Damodar Reddy	National Institute of Technology, Goa, India
Goutham Reddy	Sejong University, South Korea
N. Subba Reddy	Gyeong Sang National University, South Korea
Kazi Shah Nawaz Ripon	Norwegian University of Science and Technology, Norway
Paolo Rosso	Universitat Politècnica de València, Spain
Partha Pratim Roy	Indian Institute of Technology, Roorkee, India
Sudip Roy	Indian Institute of Technology, Roorkee, India
Sushmita Ruj	Indian Statistical Institute, Kolkata, India

Pankaj Kumar Sa	National Institute of Technology, Rourkela, India
Mounita Saha	Synopsys, India
Sanjoy Kumar Saha	Jadavpur University, Kolkata, India
Sriparna Saha	Indian Institute of Technology Patna, India
Sudipta Saha	Indian Institute of Technology, Bhubaneswar, India
Sujan Kumar Saha	Birla Institute of Technology Mesra, India
Saurav Sahay	Intel Labs, USA
Mukesh Saini	Indian Institute of Technology, Ropar, India
Ranbir Sanasam	Indian Institute of Technology, Guwahati, India
Anil Kumar Sao	Indian Institute of Technology, Mandi, India
V. Vijaya Saradhi	Indian Institute of Technology, Guwahati, India
Kamal Sarkar	Jadavpur University, Kolkata, India
Sajal Sarkar	Indian Institute of Technology, Kharagpur, India
P. S. Sastry	Indian Institute of Science, Bangalore, India
Debashish Sen	Indian Institute of Technology, Kharagpur, India
B. Uma Shankar	Indian Statistical Institute, Kolkata, India
Dipti Misra Sharma	International Institute of Information Technology, Hyderabad, India
Shirish Shevade	Indian Institute of Science, Bangalore, India
Jaya Sil	Bengal Engineering and Science University, India
Jamuna Kanta Sing	Jadavpur University, Kolkata, India
Manish Singh	Indian Institute of Technology, Hyderabad
Radu Slavescu	Technical University of Cluj Napoca, Romania
Madasamy Sornam	University of Madras, Chennai, India
P. K. Srijith	Indian Institute of Technology Hyderabad, India
Manish Srivastava	International Institute of Information Technology Hyderabad, India
Yannis Stylianou	University of Crete, Greece
Badri Narayan Subudhi	National Institute of Technology, Goa, India
Arijit Sur	Indian Institute of Technology, Guwahati, India
Shamik Sural	Indian Institute of Technology, Kharagpur, India
Tripti Swarnakar	Siksha O Anusandhan University, Bhubaneswar, India
Kumaran T.	Government Arts College for Men, Krishnagiri, India
Geetha T. V.	Anna University, Chennai, India
Sabu M. Thampi	Indian Institute of Information Technology and Management-Kerala (IIITM-K), India
Kathirvalavakumar Thangairulappan	VHNSN College, Virudhunagar, India
Veerakumar Thangaraj	National Institute of Technology, Goa, India
Srinivasan Thanukrishnan	Glosys Technology Solutions Pvt. Ltd, Chennai, India
Birjodh Tiwana	LinkedIn Inc., USA
Diana Trandabat	University Al. I. Cuza of Iasi, Romania
Turki Turki	King Abdulaziz University, Saudi Arabia
Suryakanth V. Gangashetty	International Institute of Information Technology, Hyderabad, India
Odelu Vanga	Indian Institute of Information Technology Chittoor, India

Vamsi Krishna Velidi	Indian Space Research Organization, Bangalore, India
Hrishikesh Venkataraman	Indian Institute of Information Technology Chittoor, Sricity, India
Sastry V. N.	Institute for Development and Research in Banking Technology (IDRBT)
P. Viswanath	Indian Institute of Information Technology Sricity, India
Anil Kumar Vupalla	International Institute of Information Technology Hyderabad, India
Wei Lee Woon	Masdar Institute of Science and Technology, Abu Dhabi, United Arab Emirates
Xiaolong Wu	California State University, Long Beach, USA

Additional Reviewers

Agarwal, Anurag	Das, Saurabh
Agarwal, Sheetal	Das, Sudeb
Ahmed, Arif	Dash, Ajaya Kumar
Ali, Lasker Ershad	Datta, Aloke
Alluri, Knrk Raju	De, Rajat K.
Ananthanarayanan, Rema	Dehuri, Satchidananda
Andrew, Chris	Desarkar, Maunendra Sankar
Aroyehun, Segun Taofeek	Dhall, Abhinav
Banerjee, Biplob	Dhara, Sobhan
Banerjee, Dip Sankar	Edla, Damodar Reddy
Basha, Shabbeer	Ganguly, Debasis
Basu, Tanmay	Ghosh, Kuntal
Behera, Santosh Kumar	Ghosh, Saptarshi
Behera, Shreetam	Ghosh, Sujata
Bhattacharyya, Malay	Ghosh, Susmita
Bhuyan, Sudipta	Gomez-Adorno, Helena
Biswas, Arindam	Gonuguntla, Venkateswarlu
Biswas, Saroj	Gupta, Rajeev
Biswas, Saroj K.	Jaswanth, N.
Biswas, Saroj Kr.	K. K., Santhosh
Chakraborty, Debasrita	K., Shravya
Chan, Jonathan	K., Vivekraj V.
Chanda, Bhabatosh	Karale, Vikrant
Chattopadhyay, Samiran	Kodeswaran, Palanivel
Cherukuri, Aswani Kumar	Kolagani, Nagesh
Chira, Camelia	Kolesnikova, Olga
Chowdhury, Ananda S.	Krishna, Hari
Das, Biswajit	Kumar, Nagesh
Das, Dipankar	Kummamuru, Krishna

Kundu, Ashish
 Kundu, Suman
 Law, Anwasha
 Lemnaru, Camelia
 M., Sornam
 Maheshwari, Tushar
 Maiti, Rajib Ranjan
 Maji, Pradipta
 Majumdar, Adrija
 Majumdar, Kausik
 Mamidi, Radhika
 Manchanda, Pikakshi
 Marginean, Anca
 Markov, Iliia
 Meher, Saroj K.
 Mishra, Abhijit
 Mishra, Deepak
 Mitra, Pabitra
 Mitra, Pralay
 Mitrea, Delia
 Modani, Natwar
 Mohnaty, Ramakanta
 Mondal, Ajoy
 Mukherjea, Sougata
 Murthy, K. Ramachandra
 Murthy, K. Ramachandra
 Murthy, Ramachandra
 Musti, Narasimha Murty
 Myakala, Pruthvi Raj
 N. Reganti, Aishwarya
 N., Rajasree
 Nagy, Madalina Mandy
 Nanda, P. K.
 Narayanam, Ramasuri
 Nasipuri, Mita
 Nath, Bhabesh
 Negi, Atul
 Nekuri, Naveen
 Nekuri, Naveen
 Nigam, Aditya
 P. Y. K. L., Srinivas
 Pal, Sukomal
 Pamulapati, Trinadh Reddy
 Pillai, Gargi V.
 Podila, Sahithi
 Pokharel, Shiraj
 Pratihar, Dilip
 R, K
 Raju, K. Srinivasa
 Ravi, Kumar
 Reddy, Damodar
 Reddy, Goutham
 Ripon, Kazi Shah Nawaz
 Roy, Abhinaba
 Roy, Partha Pratim
 Roy, Rahul
 Roy, Sudip
 Ruj, Sushmita
 Sa, Pankaj K.
 Sadhukhan, Payel
 Saha, Sanjoy Kumar
 Saha, Sujan
 Saini, Mukesh
 Sanasam, Ranbir
 Sarkar, Kamal
 Sengupta, Debanjan
 Shankar, B. Uma
 Sharma, Shakti
 Sil, Jaya
 Sing, Jamuna Kanta
 Singh, Manish
 Sk, Arif Ahmed
 Slavescu, Radu Razvan
 Subudhi, Badri Narayan
 Sur, Arijit
 Surampudi, Bapi Raju
 Swarnkar, Tripti
 T., Kumaran
 Thangaraj, Veerakumar
 Thirumuru, Rama Krishna
 Tiwana, Birjodh
 Vamsi, Vallurupalli
 Vara Prasad, Raja
 Vegesna, Vishnu Vidyadhara Raju
 Verma, Ashish
 Vijaya, Saradhi

Contents

Functional Link Artificial Neural Network for Multi-label Classification	1
<i>Anwasha Law, Konika Chakraborty, and Ashish Ghosh</i>	
Emotion Recognition Through Facial Gestures - A Deep Learning Approach	11
<i>Shrija Mishra, Geeta Ramani Bala Prasada, Ravi Kant Kumar, and Goutam Sanyal</i>	
Supervised Approaches to Assign Cooperative Patent Classification (CPC) Codes to Patents	22
<i>Tung Tran and Ramakanth Kavuluru</i>	
A Betweenness Centrality Guided Clustering Algorithm and Its Applications to Cancer Diagnosis	35
<i>R. Jothi</i>	
MahalCUSFilter: A Hybrid Undersampling Method to Improve the Minority Classification Rate of Imbalanced Datasets	43
<i>Venkata Krishnaveni Chennuru and Sobha Rani Timmappareddy</i>	
Bezier Curve Based Continuous Medial Representation for Shape Analysis: A Theoretical Framework.	54
<i>Leonid Mestekiy and B. H. Shekar</i>	
Trust Distrust Enhanced Recommendations Using an Effective Similarity Measure	64
<i>Stuti Chug, Vibhor Kant, and Mukesh Jadon</i>	
Language Identification Based on the Variations in Intonation Using Multi-classifier Systems	73
<i>Shinjini Ghosh</i>	
Cognitive Decision Making for Navigation Assistance Based on Intent Recognition.	81
<i>Sumant Pushp, Basant Bhardwaj, and Shyamanta M. Hazarika</i>	
Clinical Intelligence: A Data Mining Study on Corneal Transplantation	90
<i>Brian Carneiro, Rui Peixoto, Filipe Portela, and Manuel Filipe Santos</i>	
High-Quality Medical Image Compression Using Discrete Orthogonal Cosine Stockwell Transform and Optimal Integer Bit Allocated Quantization	100
<i>Vikrant Singh Thakur, Kavita Thakur, and Shubhrata Gupta</i>	

Coprime Mapping Transformation for Protected and Revocable Fingerprint Template Generation	111
<i>Rudresh Dwivedi and Somnath Dey</i>	
Supervised Asymmetric Metric Extraction: An Approach to Combine Distances	123
<i>Archil Maysuradze, B. H. Shekar, and Mikhail Suvorov</i>	
Interval-Valued Writer-Dependent Global Features for Off-line Signature Verification	133
<i>K. S. Manjunatha, D. S. Guru, and H. Annapurna</i>	
Despeckling with Structure Preservation in Clinical Ultrasound Images Using Historical Edge Information Weighted Regularizer	144
<i>Rahul Roy, Susmita Ghosh, Sung-Bae Cho, and Ashish Ghosh</i>	
Fingerprint Image Quality Assessment and Scoring	156
<i>Ram Prakash Sharma and Somnath Dey</i>	
A Multi-objective Evolutionary Algorithm for Color Image Segmentation	168
<i>Kazi Shah Nawaz Ripon, Lasker Ershad Ali, Sarfaraz Newaz, and Jinwen Ma</i>	
Face Recognition by RBF with Wavelet, DCV and Modified LBP Operator Face Representation Methods	178
<i>J. Jebakumari Beulah Vasanthi and T. Kathirvalavakumar</i>	
DNN-HMM Acoustic Modeling for Large Vocabulary Telugu Speech Recognition	189
<i>Vishnu Vidyadhara Raju Vegesna, Krishna Gurugubelli, Hari Krishna Vydana, Bhargav Pulugandla, Manish Shrivastava, and Anil Kumar Vuppala</i>	
Memetic Algorithm Based on Global-Best Harmony Search and Hill Climbing for Part of Speech Tagging	198
<i>Luz Marina Sierra Martínez, Carlos Alberto Cobos, and Juan Carlos Corrales</i>	
A Study on Crossmodal Correspondence in Sensory Pathways Through Forced Choice Task and Frequency Based Correlation in Sound-Symbolism	212
<i>Keerthi S Chandran, Swati Banerjee, and Kuntal Ghosh</i>	
Point Process Modeling of Spectral Peaks for Low Resource Robust Speech Recognition	221
<i>Anupam Mandal, K. R. Prasanna Kumar, and Pabitra Mitra</i>	

Significance of DNN-AM for Multimodal Sentiment Analysis	231
<i>Harika Abburi, Rajendra Prasath, Manish Shrivastava, and Suryakanth V. Gangashetty</i>	
Pattern Based Information Retrieval Approach to Discover Extremist Information on the Internet.	240
<i>Mikhail Petrovskiy, Dmitry Tsarev, and Irina Pospelova</i>	
A Concept Driven Graph Based Approach for Estimating the Focus Time of a Document	250
<i>Shashank Shrivastava, Mitesh Khapra, and Sutanu Chakraborti</i>	
Query Morphing: A Proximity-Based Approach for Data Exploration and Query Reformulation.	261
<i>Jay Patel and Vikram Singh</i>	
WikiSeeAlso: Suggesting Tangentially Related Concepts (<i>See also links</i>) for Wikipedia Articles	274
<i>Sahiti Labhishetty, Ayesha Siddiqa, Rajivteja Nagipogu, and Sutanu Chakraborti</i>	
Integrating Knowledge Encoded by Linguistic Phenomena of Indian Languages with Neural Machine Translation.	287
<i>Ruchit Agrawal, Mihir Shekhar, and Dipti Misra</i>	
Partitioned-Based Clustering Approaches for Single Document Extractive Text Summarization	297
<i>Prannoy Subba, Susmita Ghosh, and Rahul Roy</i>	
Arousal Prediction of News Articles in Social Media.	308
<i>Nagendra Kumar, Anusha Yadandla, K. Suryamukhi, Neha Ranabothu, Sravani Boya, and Manish Singh</i>	
Sentiment Analysis of Tweets in Malayalam Using Long Short-Term Memory Units and Convolutional Neural Nets	320
<i>S. Sachin Kumar, M. Anand Kumar, and K. P. Soman</i>	
Improved Community Interaction Through Context Based Citation Analysis.	335
<i>Baishali Saha, Tanushree Anand, Anurag Sharma, and Bibhas Ghoshal</i>	
Mining Informative Words from the Tweets for Detecting the Resources During Disaster.	348
<i>Madichetty Sreenivasulu and M. Sridevi</i>	
An Ensemble Based Method for Predicting Emotion Intensity of Tweets	359
<i>Sreekanth Madisetty and Maunendra Sankar Desarkar</i>	

A Graph-Based Frequent Sequence Mining Approach
to Text Compression 371
C. Oswald, I. Ajith Kumar, J. Avinash, and B. Sivaselvan

ULR-Discr: A New Unsupervised Approach for Discretization 381
Habiba Drias, Nourelhouda Rehkab, and Hadjer Moulai

Identifying Terrorist Index (T^+) for Ranking Homogeneous Twitter Users
and Groups by Employing Citation Parameters and Vulnerability Lexicon . . . 391
Soumyadeep Debnath, Dipankar Das, and Bappaditya Das

Soft Metaphor Detection Using Fuzzy c-Means. 402
Sunny Rai, Shampa Chakraverty, Devendra K. Tayal, and Yash Kukreti

A Study on CART Based on Maximum Probabilistic-Based Rough Set 412
Utpal Pal, Sharmistha Bhattacharya (Halder), and Kalyani Debnath

Portfolio Optimization in Dynamic Environments Using MemSPEAII 424
Priyank Shah and Sanket Shah

Author Index 437

Functional Link Artificial Neural Network for Multi-label Classification

Anwasha Law¹, Konika Chakraborty², and Ashish Ghosh¹(✉)

¹ Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700108, India
{anweshalaw_r,ash}@isical.ac.in

² Department of Computer Science, Vidyasagar University, Medinipur 721102, India
konika.chakraborty1992@gmail.com

Abstract. In this article, a multi-label functional link artificial neural network (MLFLANN) has been developed to efficiently perform multi-label data classification. The input data is functionally expanded to a higher dimension, followed by iterative learning of the multi-label FLANN (MLFLANN) using the training set. The architecture of the network is less complex and the input space dimension is improved in an attempt to overcome the non-linear nature of the multi-label classification problem. The method has been validated on various multi-label datasets and the results are found to be encouraging.

Keywords: Multi-label classification · Neural networks
Functional link artificial neural networks

1 Introduction

Multi-label classification [10] is a part of machine learning that deals with data which can belong to more than one classes at the same time. For example, while categorizing news articles it might be seen that one particular article is about a charity football match among movie actors. From traditional classification perspective, there is an ambiguity whether the article should be categorized under ‘sports’ or ‘entertainment’. From a multi-label classification view, the news article can be classified as both. Similarly, a movie, song or novel might belong to more than one genres. If these types of data are forcefully put into any one of the classes, it ignores the information in the data that might prominently describe the other classes. Thus, multi-label classification helps to deal with ambiguity of data and its main aim is to predict a set of classes for any unknown data. In the past decade, the field of multi-label classification has been explored and some amount of research work has been done in the domains of text categorization [18], labeling of multi-media data [1], etc. There are mainly two ways of handling multi-label classification - *data transformation* and *problem adaptation* [10]. The data transformation approach [1] handles the multi-label property mainly by converting the data to a simpler (binary/single-label) representation. However, this approach was used previously, and is not frequently availed in recent times.

The more popular approach is that of problem adaptation [18]. This group of techniques do not modify the original multi-label data, instead they adapt an existing method to efficiently perform multi-label classification. There are several adaptation-based classifiers that have been explored by researchers. The techniques can be broadly categorized into – neural networks [18], support vector machines [8], instance-based [19], tree-based [4] and probabilistic methods [2].

Neural networks [9] are one of the most popular tools in machine learning and have been explored in recent times to handle multi-label classification tasks. Multi-label data have complex class boundaries which are quite difficult to detect, which makes neural networks quite suitable for classification of this kind of data. Applications of different types of neural networks like multilayer perceptron (MLP) [18], radial basis networks (RBF) [17], extreme learning machines (ELM) [11], etc. have already been explored in the field of multi-label data classification. Some of the existing models are computationally expensive due to their complex architecture which is required for adapting to non-linearity in data. A few of the networks are not able to efficiently classify multi-label data due to a very simplistic approach.

Keeping these drawbacks in mind, we propose a simple yet efficient architecture – a multi-label functional link artificial neural network (MLFLANN) for the class prediction of multi-label data. This model has been adapted from the FLANN for single-label data classification [12]. The input layer of the network incorporates a higher dimension projection of the features which make the output space more discriminated, thus leading to efficient classification. The output layer has been modified to contain multiple output neurons (one for each class) as opposed to the single output neuron in the existing FLANN. Our proposed model is structurally quite simple, thus involving less computational complexity. As per the knowledge of the authors, there has been little or no reference of functional link ANNs in the literature for multi-label data classification. This motivated us to pursue the present study of developing a multi-label functional link ANN (MLFLANN). Our proposed model has been validated with four datasets and compared against two other existing algorithms.

In the next section some existing methods of multi-label classification and their shortcomings are discussed. Section 3 has some preliminary description of multi-label data and the existing FLANN. Section 4 is an elaboration of the proposed work, with detailed architecture, training and testing phases. Section 5 discusses the results and compares MLFLANN with few other existing techniques. The last Sect. 6 concludes the paper.

2 Related Works

Multi-label classification has been explored by researchers mostly in the last decade. As mentioned earlier, there are mainly two approaches that have been used for multi-label data classification, namely, data transformation and method adaptation approaches. There are several data transformation approaches that exist in the literature, like Model-s, Model-i, Model-n [1], etc. These approaches

first convert the multi-label data to binary/single-label data, and then use an existing classifier. These techniques either lead to loss of information which eventually increases misclassification, or they include a lot of redundancy in the data, thus unnecessarily increasing the computation complexity.

On the contrary, adaptation-based techniques modify traditional classifiers in a way that it can tackle the original multi-label data without tampering with it. Most of the method adaptation approaches rely on traditional algorithms based on trees [4], neural networks [18], instance-based learning [19], etc. Among the instance-based multi-label classification algorithms that exist in literature, ML-KNN [19] is one of the best known algorithm. It internally works as a binary relevance classifier, since a separate set of apriori and conditional probabilities are independently computed for each label. It involves a large amount of computations in the second order neighbourhood of each training pattern before the actual classification is done. For a dataset with huge number of samples, this method would be computationally expensive and quite slow.

Apart from these, there are a few works based on artificial neural networks (ANN); the first multi-label adapted ANN is BP-MLL [18]. It is a simple two-layer neural network model which uses back propagation algorithm to train itself. The appropriate number of hidden neurons needed in that single hidden layer is found experimentally. This poses a problem since, different datasets may require different number of hidden neurons.

ML-RBF [17] is an adaptation of radial basis networks for ML classification. It executes K-means clustering and uses the cluster centers in the RBF model. There is no way to know the distribution of the dataset beforehand, thus, K-means may not always cluster the data well. The two-step process also increases the computational complexity of the problem.

Multi-label Extreme Learning Machine (ML-ELM) algorithm in [11] uses an ELM network with one hidden layer. Since the mechanism incorporates randomness and the learning takes place in one pass, it needs a very large number of hidden neurons (found experimentally) in comparison to the number of features. Hence, this technique is not suitable for classification of complex data with large number of features.

From the existing works, it is seen that neural networks have been used in the classification of multi-label data, but there are a few shortcomings that need to be handled. Our proposed approach, MLFLANN, has a simple architecture that learns iteratively thus is computationally cheaper yet efficient. It also incorporates functional expansion of the input space to handle the non-linearity effectively. In this way, the proposed model attempts to handle few of the drawbacks faced in the existing works and perform multi-label classification effectively.

3 Preliminaries

In this section, a brief overview of the FLANN and multi-label data have been provided.

3.1 Functional Link Artificial Neural Network (FLANN)

Neural networks are widely used to handle complex classification problems. Various models of ANNs have been used in the past to solve different types of problems. Functional link artificial neural network (FLANN) is one such neural network model, which is simple yet efficient and has been used to solve classification tasks. FLANN is a flat feed-forward neural network with a functionally expanded input layer, no hidden layers and an output layer with one neuron. It follows a simple learning rule and uses the single error generated by the network to train itself iteratively. Its low architectural complexity makes it easier to train and helps to gain more insight into the classification problem. FLANN uses functionally expanded features to increase the dimensionality of the input data, thus overcoming the non-linear nature of the given problem. From Cover’s theorem [3], it is known that given a set of training data that is not linearly separable, one can with high probability transform it into a training set that is linearly separable by projecting it into a higher-dimensional space via some non-linear transformation. Hence, the hyper-planes that are generated by FLANN should be able to efficiently discriminate between the input patterns.

In literature, it is seen that FLANN has been modeled for data classification. In [13, 14] it has been shown that functional links neurons may be conveniently used for function approximation. These models have lesser computational load and faster convergence rate than multi-layer perceptron. In [12], FLANN was used with gradient descent method for classification task of data mining where a different set of orthonormal basis functions was suggested for feature expansion. Further, FLANN based classifiers have been combined with genetic algorithms in [5, 6] and PSO in [7] for enhancing the classification accuracy.

Exploring the various domains and variations of FLANN, it was seen that this network has proven to be quite efficient in single-label classification tasks. The major characteristic of FLANN by which it projects the input vector efficiently to a higher dimension to improve separability makes it quite suitable for multi-label data. The class boundaries of multi-label datasets are inevitably overlapped and the data eventually seems to be quite difficult to classify. Projecting the input vectors to a higher dimension might make this problem comparatively simpler. As per the knowledge of the authors, FLANN has not been adapted previously for multi-label classification. With this motivation, we propose a multi-label functional link artificial neural network in Sect. 4. Before moving on to the proposed work, a brief description of multi-label data is provided for better understanding.

3.2 Representation of Multi-label Data

If a multi-label input dataset contains N data points, the i^{th} input pattern is represented as a feature vector $\mathbf{X}_i = \{x_{i1}, x_{i2}, \dots, x_{id}\}$. Each element x_{ij} is a feature, where $1 \leq j \leq d$ and d is the dimension of the input space. Each of these input patterns \mathbf{X}_i is associated with a corresponding label set vector $\mathbf{Y}_i = \{y_{i1}, y_{i2}, \dots, y_{iC}\}$, where C is the dimension of the output space. Unlike

traditional single label datasets, where any one of these C labels is assigned to 1 and rest to 0, a multi-label data have multiple 1's in the label set. $y_{ic} = 1$ indicates that the i^{th} input pattern belongs to the c^{th} class, therefore it is relevant, and $y_{ic} = 0$ indicates an irrelevant label. These target values are kept fixed during the training and testing phases.

4 The Proposed Multi-label FLANN (MLFLANN)

In the present work, an adaptation of functional link artificial neural network has been proposed for class prediction of multi-label data where every pattern may belong to more than one class at a time. Multi-label data has overlapping class boundaries, hence, the output space of this kind of data is quite complex. The functional expansion of features in MLFLANN helps to generate hyperplanes that have a higher discrimination capability suitable for multi-label data.

4.1 Architecture of the Network

Architecture of the existing FLANN has been modified in our proposed work to incorporate classification of multi-label data (Fig. 1). The basic feed-forward network model has two layers, the expanded input layer and the output layer. The actual input to the network has d features. Each input feature x_{ij} is functionally expanded as $\{f_1(x_{ij}), f_2(x_{ij}), \dots, f_P(x_{ij})\}$, where P is the total number

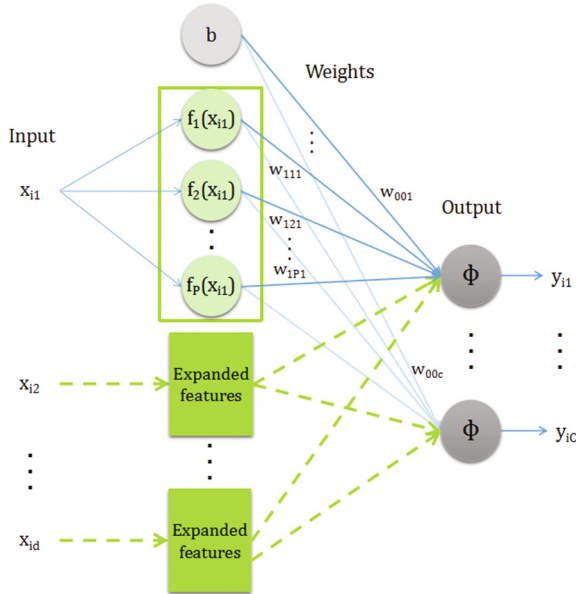


Fig. 1. Architecture of MLFLANN

of basis functions used for each input element. This increases the dimension of the input space from d to $P*d$, by a set of basis functions F applied on an input pattern X_i . $F(X_i)$ can be expanded as

$$F(X_i) = \{f_1(x_{i1}), f_2(x_{i1}), \dots, f_p(x_{i1}), \\ f_1(x_{i2}), f_2(x_{i2}), \dots, f_p(x_{i2}), \dots, \\ f_1(x_{id}), f_2(x_{id}), \dots, f_p(x_{id})\}. \quad (1)$$

Once the new expanded features are obtained from the input data, they are then fed to the network. Since, there are no hidden layers, the other layer in the network is the output layer. Unlike the existing FLANN which has only one output node, the proposed MLFLANN has C output neurons, one for each class. Since the existing FLANN architecture generates only one global error, representation of multiple outputs and learning multiple labels is not possible using the earlier network architecture. Hence, the adaptation is necessary. The proposed model is a feed-forward network, with $d*P*C$ number of connections between the expanded input layer and the output layer.

4.2 Training Phase

In the proposed MLFLANN architecture, a set of basis functions F , and a fixed number of weight parameters W have been used to represent the output \mathbf{Y} . The output of multi-label data can be represented as a vector of individual class outputs, i.e., $\mathbf{Y}_i = \{y_{i1}, y_{i2}, \dots, y_{ic}\}$. With a specific set of basis functions F , the challenge is to find the weight parameters W that provide the best possible approximation of \mathbf{Y} on the given input-output samples. This can be achieved by iteratively updating W .

At the beginning of the training phase, the network weights are initialized randomly. Then, the input patterns \mathbf{X}_i are fed to the MLFLANN one at a time. Each of the input features are functionally expanded; some trigonometric basis functions have been used in our problem to expand an input feature x_{ij} :

$$F(X_i) = \{\sin \pi(x_{ij}), \cos \pi(x_{ij}), \\ \sin 2\pi(x_{ij}), \cos 2\pi(x_{ij}), \\ \dots, \\ \sin m\pi(x_{ij}), \cos m\pi(x_{ij})\}. \quad (2)$$

A weighted sum of these nonlinear outputs are computed through the network. The induced local fields for each class is obtained by adding a bias b to this sum. An activation function ϕ is applied at each of the output nodes to obtain the estimated outcomes for all the classes. The output vector \mathbf{Y}'_i can be written as,

$$\mathbf{Y}'_i = \{y'_{i1}, y'_{i2}, \dots, y'_{iC}\}. \quad (3)$$

$$y'_{ic} = \phi \left(\sum_{j=1}^d \sum_{p=1}^P f_p(x_{ij}) \cdot w_{jpc} + b \right), \quad 1 \leq c \leq C. \quad (4)$$

This actual output \mathbf{Y}'_i is compared to the corresponding desired output \mathbf{Y}_i and the resultant error vector \mathbf{E} for the i^{th} pattern is

$$\begin{aligned} \mathbf{E} &= \mathbf{Y}_i - \mathbf{Y}'_i \\ &= \{y_{i1} - y'_{i1}, y_{i2} - y'_{i2}, \dots, y_{iC} - y'_{iC}\} \\ &= \{e_1, e_2, \dots, e_C\}. \end{aligned} \quad (5)$$

At the $t + 1^{th}$ iteration, the weight matrix W is updated depending on the error computed at the t^{th} iteration. The change in weight matrix in the t^{th} iteration, $\Delta W^{(t)}$, is a set of weight vectors $\Delta \mathbf{w}_{pj}^{(t)}$ given as,

$$\Delta \mathbf{w}_{pj}^{(t)} = \mu \cdot f_p(x_j)^{(t)} \cdot \boldsymbol{\delta}^{(t)}, \quad (6)$$

where, μ is the learning rate, $f_p(x_j)^{(t)}$ is the expanded input feature x_j by function f_p at the t^{th} iteration and the corresponding gradient vector is

$$\boldsymbol{\delta}^{(t)} = \mathbf{Y}' \cdot (1 - \mathbf{Y}') \cdot \mathbf{E}, \quad (7)$$

where, \mathbf{Y}' is the output vector and \mathbf{E} error vector at output layer. Then the connection weights for the $t + 1^{th}$ iteration can be updated as

$$W^{(t+1)} = W^{(t)} + \Delta W^{(t)}, \quad (8)$$

where, $W^{(t)}$ is the weight at the t^{th} iteration.

At the end of the training phase, the learned classifier is able to generate a set of outputs for a given pattern, but this output vector needs to be mapped to a label set. In the case of multi-label data, we need to determine a suitable threshold which will be able to correctly map all the class labels. Either a global threshold can be fixed, or a set of thresholds may be determined, one for each output node. To maintain simplicity of our model, a global threshold had been determined experimentally and has been used in the testing phase as well.

4.3 Testing Phase

Once the MLFLANN is well trained, the validation/testing phase begins. In this phase, each multi-label test pattern is taken at a time and fed to the trained MLFLANN. The same set of basis functions are used to expand the features of the test pattern. The trained network computes the output at each node corresponding to each class. Once the outputs have been obtained, the global threshold is applied to the output vector. The output obtained after thresholding gives the actual set of class labels for the test pattern.

5 Experimental Details and Analysis of Results

To evaluate the effectiveness of the proposed MLFLANN algorithm, it has been validated over four datasets. In the following sections, details of the datasets used and the corresponding results obtained have been discussed.

5.1 Datasets Used

The proposed approach has been evaluated using four multi-label datasets, namely scene (image) [1], yeast (gene) [8], emotions (audio) [15] and CAL500 (audio) [16]. These are few of the common datasets that are used by researchers to validate multi-label classification methods. Preprocessed versions of the datasets are freely available at <http://mulan.sourceforge.net/datasets-mlc.html>. For each dataset, the features have been preprocessed and represented numerically, and the output is given as a set of class labels, i.e., the values 0 or 1 are assigned for each irrelevant or relevant class respectively. The feature values of the input patterns have been normalized between 0 to 1 before experimentation.

5.2 Results and Analysis

To assess the performance of the proposed approach, experiments have been conducted on four multi-label datasets and has been compared to two other neural network based multi-label classification algorithms. One is a multi-label single-layer perceptron model which works without modifying the input features. The other model used for comparison is ML-RBF [17], this technique modifies the input features using radial basis functions. Results corresponding to the above two techniques and our proposed work MLFLANN have been shown in Table 1.

Table 1. Comparative results on four datasets

Dataset	Method	Hamming loss ↓	Average precision ↑	Ranking loss ↓	Coverage ↓	One error ↓
Scene	MLSLP	0.1638	0.5884	0.2941	1.5627	0.6149
	MLRBF	0.0771	0.8877	0.1678	0.4299	0.1834
	MLFLANN	0.1185	0.8251	0.1106	0.6476	0.2794
Yeast	MLSLP	0.2378	0.7076	0.2242	7.4033	0.2521
	MLRBF	0.2361	0.7278	0.2228	7.1468	0.2105
	MLFLANN	0.2246	0.7377	0.2129	7.0105	0.1898
Emotions	MLSLP	0.2211	0.7873	0.1928	2.0169	0.2798
	MLRBF	0.1936	0.7961	0.1616	1.8233	0.2583
	MLFLANN	0.2031	0.7983	0.1607	1.8099	0.2677
CAL500	MLSLP	0.2867	0.3297	0.4603	169.3410	0.2980
	MLRBF	0.3045	0.2435	0.4725	168.5221	0.3146
	MLFLANN	0.1872	0.4152	0.2678	154.7961	0.1245

As performance measuring indices, hamming loss, average precision, ranking loss, coverage and one error have been used [10]. To compute an average performance, k-fold cross validation technique has been chosen. From the results

obtained, it is seen that the proposed method performs better than the single layer perceptron model for all the datasets. It is safe to say that the functional expansion in our proposed model has made the multi-label data more discriminable. On the other hand, MLFLANN and MLRBF have comparable performance. Both the techniques do not use the input features as it is, and perform some transformation of the input space to make the classification task simpler. MLFLANN performs better than MLRBF for yeast and CAL500 datasets, and is a close competitor for the other two datasets. The results strengthen the fact that functional expansion of features and iteratively adapting the network weights in MLFLANN has proven to be beneficial for classification of multi-label data.

6 Conclusion

This article presents a multi-label functional link ANN for the class prediction of multi-label datasets. FLANN is a neural network architecture which has been previously used by various researchers to perform efficient classification. However, this neural model had never been explored in the context of multi-label data classification. The MLFLANN expands the input data to a higher dimension which helps to classify the multi-label data well. This type of data is quite complex and has overlapping class boundaries which makes multi-label classification more challenging than single-label classification tasks. The proposed algorithm was tested on various datasets and compared with other multi-label classification techniques with encouraging results. Future work includes modifying the proposed model to incorporate an adaptive learning strategy which can be more useful for multi-label data classification.

References

1. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. *Pattern Recogn.* **37**(9), 1757–1771 (2004)
2. Cheng, W., Hüllermeier, E., Dembczynski, K.J.: Bayes optimal multilabel classification via probabilistic classifier chains. In: *Proceedings of the 27th International Conference on Machine Learning, ICML 2010*, pp. 279–286 (2010)
3. Cover, T.M.: Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Trans. Electron. Comput.* **3**, 326–334 (1965)
4. De Comit e, F., Gilleron, R., Tommasi, M.: Learning multi-label alternating decision trees from texts and data. In: Perner, P., Rosenfeld, A. (eds.) *MLDM 2003*. LNCS, vol. 2734, pp. 35–49. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-45065-3_4
5. Dehuri, S., Cho, S.B.: Evolutionarily optimized features in functional link neural network for classification. *Expert Syst. Appl.* **37**(6), 4379–4391 (2010)
6. Dehuri, S., Cho, S.B.: A hybrid genetic based functional link artificial neural network with a statistical comparison of classifiers over multiple datasets. *Neural Comput. Appl.* **19**(2), 317–328 (2010)

7. Dehuri, S., Roy, R., Cho, S.B., Ghosh, A.: An improved swarm optimized functional link artificial neural network (ISO-FLANN) for classification. *J. Syst. Softw.* **85**(6), 1333–1345 (2012)
8. Elisseff, A., Weston, J.: A kernel method for multi-labelled classification. In: *Advances in Neural Information Processing Systems*, pp. 681–687 (2002)
9. Haykin, S.: *Neural Networks: A Comprehensive Foundation*. Prentice - Hall of India, New Delhi (2008)
10. Herrera, F., Charte, F., Rivera, A.J., Del Jesus, M.J.: *Multilabel Classification: Problem Analysis, Metrics and Techniques*. Springer, Cham (2016). <https://doi.org/10.1007/978-3-319-41111-8>
11. Kongsorot, Y., Horata, P.: Multi-label classification with extreme learning machine. In: *6th International Conference on Knowledge and Smart Technology (KST)*, pp. 81–86. IEEE (2014)
12. Misra, B.B., Dehuri, S.: Functional link artificial neural network for classification task in data mining. *J. Comput. Sci.* **3**, 948–955 (2007)
13. Pao, Y.-H.: *Adaptive Pattern Recognition and Neural Networks*. Addison-Wesley, Boston (1989)
14. Pao, Y.-H., Phillips, S.M., Sobajic, D.J.: Neural-net computing and the intelligent control of systems. *Int. J. Control* **56**(2), 263–289 (1992)
15. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.P.: Multi-label classification of music into emotions. In: *Ninth International Conference on Music Information Retrieval (ISMIR)*, vol. 8, pp. 325–330 (2008)
16. Turnbull, D., Barrington, L., Torres, D., Lanckriet, G.: Semantic annotation and retrieval of music and sound effects. *IEEE Trans. Audio Speech Lang. Process.* **16**(2), 467–476 (2008)
17. Zhang, M.L.: ML-RBF: RBF neural networks for multi-label learning. *Neural Process. Lett.* **29**(2), 61–74 (2009)
18. Zhang, M.L., Zhou, Z.H.: Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Trans. Knowl. Data Eng.* **18**(10), 1338–1351 (2006)
19. Zhang, M.L., Zhou, Z.H.: ML-KNN: a lazy learning approach to multi-label learning. *Pattern Recogn.* **40**(7), 2038–2048 (2007)

Emotion Recognition Through Facial Gestures - A Deep Learning Approach

Shrija Mishra^(✉), Geeta Ramani Bala Prasada^(✉), Ravi Kant Kumar^(✉),
and Goutam Sanyal^(✉)

Department of Computer Science and Engineering, National Institute of Technology Durgapur,
Durgapur, India

shrija.mishra102@gmail.com, geetabala9@gmail.com,
vit.ravikant@gmail.com, nitgsanyal@gmail.com

Abstract. As defined by some theorists, human emotions are discrete and consistent responses to internal or external events which have significance for an organism. They constitute a major part of our non-verbal communication. Among the human emotions, happy, sad, fear, anger, surprise, disgust and neutral are the seven basic emotions. Facial expressions are the best way to exhibit emotions. In this era of booming human-computer interaction, enabling the machines to recognize these emotions is a paramount task. There is an amalgamation of emotions in every facial expression. In this paper, we identified the different emotions and their intensity level in a human face by implementing deep learning approach through our proposed Convolution Neural Network (CNN). The architecture and the algorithm here yield appreciable results that can be used as a motivation for further research in computer based emotion recognition system.

Keywords: Face detection · Emotion recognition · Human-computer interaction
Convolutional Neural Network (CNN) · Deep learning · Cross validation · SVM

1 Introduction

Communication plays a key role in our daily lives. In this era of technology, human-computer interaction (HCI) and automation, emotional recognition has become an indispensable field of study. Facial expressions and our actions are non-verbal means of communication which comprise of 93% human communication, of which facial gestures and human actions have 55% role [1]. Facial expressions are universal and important in establishing interpersonal relations. There are seven basic emotions [2]. These include happy, sad, anger, disgust, surprise, fear and neutral. All other emotions are a result of the heterogeneity of these emotions.

Some significant contributions made in this area are Facial expression recognition based on Local Binary Patterns [3]. Emotion recognition using binary decision tree [4], Facial Expression Recognition with Convolutional Neural Networks [5]. Modular Eigen spaces method for emotion classification using NN and HMM [6], Emotion analysis in visual and audio cues [7], Combining multiple kernel methods [8]. But these computational methods have far behind than human accuracy as their foundation is not based on the functioning of human deep learning and training.

The objective of our research is to examine the facial emotion in static images using various attempted Convolutional Neural Network (CNN). CNN [9] is a special kind of deep learning method that provides solutions to many problems in image recognition after huge training. Due to lack of large amount of training it is difficult even for the humans to detect an emotion in a face. For example, we cannot absolutely determine whether a person is surprised or happy. Thus, we try to delve into the matter and analyze different level of emotions present in a human face at an instance. FER-2013 [10] database present these emotions into 7 categories Neutral, Happy, Sad, Surprise, Disgust, Fear, Anger. Accuracy of 63.03% was obtained on absolute classifications. For the ambiguous emotions, considering the top 2 results as correct, we achieved an accuracy of 67%. To improve the performance, we applied regularizations, dropout, batch normalization using grid search and transfer learning.

Further, the paper is divided into 10 sections. Section 2 describes the dataset. In Sect. 3, pre-processing task has been applied on the dataset. Section 4 comprises SVM. Overall architecture of our proposed system has been mentioned in Sect. 5. CNN and our proposed network are discussed in Sects. 6 and 7 respectively. In Sect. 8, finally selected proposed network has been described. Section 9 comprises of emotions results and finally, Sect. 10 draws the concluding remarks.

2 Dataset

FER2013 dataset [10] has been used for the experiment. It consists of 37887 pre-cropped gray scale images with size of $48 * 48$. The images are labeled in 7 emotions (0 = Anger, 1 = Disgust, 2 = Fear, 3 = Happy, 4 = Sad, 5 = Surprise, 6 = Neutral) (Fig. 1).



Fig. 1. FER2013 dataset sample

3 Preprocessing

In general scenario, human vision system, first detects the faces, and then subsequently it recognizes the emotion associated with that face. In the same way, in this work, face detection is the pre-processing or prior work of the emotion recognition task. Face detection task has been done using Viola Jones algorithm [11] (Fig. 2).

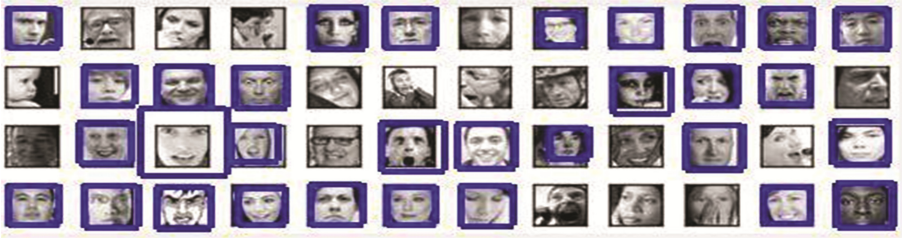


Fig. 2. Detected faces using Viola Jones algorithm

Haar Feature-Based Cascaded Classifier [11] is applied on all the images. This forms a bounding box around the face in the images. The area inside the bounding box is cropped and reshaped into 48×48 pixels. After pre-processing, the dataset consists of 11,246 images of the 7 emotions of which 1456 are angry, 240 are disgust, 1414 fear, 3235 happy, 1304 sad, 1362 surprise and 2235 are neutral. All the images are of frontal face. Non frontal faces (image of side face) and non-relevant images (images that were some random image or those with hands covering face, etc.) were removed (Fig. 3).

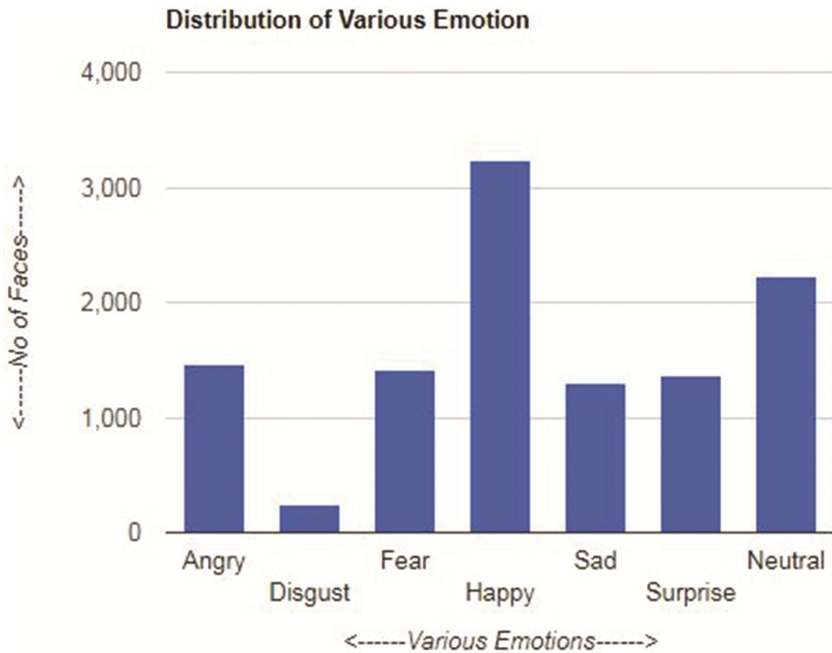


Fig. 3. Distribution of emotions after pre-processing

4 Emotion Prediction Using SVM

First, we applied SVM [12], previously the best-known image classification technique for testing its efficiency in the work of emotion detection. It is a supervised learning classification method that relies on results from statistical learning theory to guarantee high generalization performance. They are non-parametric models that need proper parameter tuning. The complexity and the computational cost grow with the number of training samples and the number of classes. The pre-processed images were used for training and testing. This multi class classification was carried out using the SVC function of scikit learn library. Training was performed on 70% of the data and the rest was used for testing. An accuracy of only 46.74% was attained on the test data.

To try out a different method and for better performance, we went on to deep learning that is the most trending area of research and application in this era as it is known to give the best results to complex problems such as image classification, natural language processing, and speech recognition.

5 System Architecture

For better understanding, overall architecture of our proposed work has been divided into two phases. They have been termed as training and testing phase (Fig. 4).



Fig. 4. System architecture

5.1 Training Phase

The proposed network has been trained with about 7800 training images (70% of the images after pre-processing) taken from FER-2013 [10] database. This database contains seven standard categories of emotions for every subject. During training process deep convolutional neural network has been applied for feature extraction and training classification. It uses supervised learning approach over huge number of images. The proposed convolutional neural network (CNN) has been explained in Sect. 8 (Fig. 5).

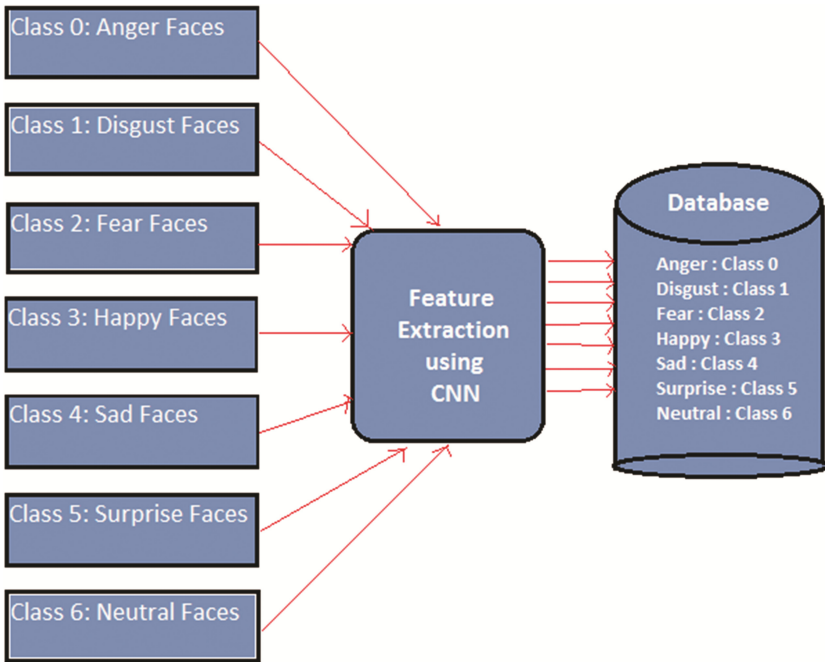


Fig. 5. Proposed training architecture

5.2 Testing Phase

The proposed network has been tested with about 3400 test images (30% of the images after pre-processing) taken from FER-2013 database. Like training phase, in the testing

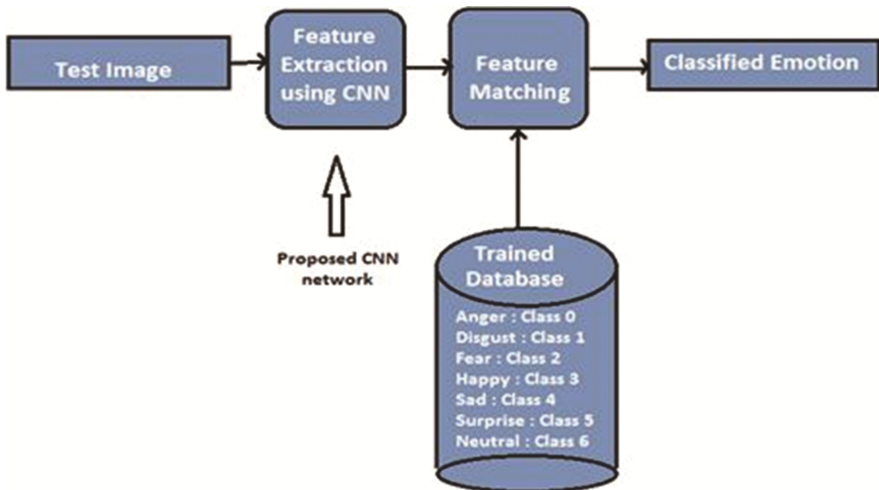


Fig. 6. Proposed testing architecture

phase, feature extraction is completed using proposed convolutional neural network. But classification of emotion is decided after matching of extracted features with trained features (Fig. 6).

6 Convolutional Neural Network

Convolutional neural networks have the most influential innovations in the field of computer vision. It is biologically inspired from visual cortex and imitates the working of human brain for visual analysis.

All the networks described here are programmed using Keras, a deep learning python library on Tensorflow in the backend. This facilitated faster and easier experimentation. ConvNet architectures make the explicit assumption that the inputs are images, which allows us to encode certain properties into the architecture. The image is fed into the network and then the network analyses the features of the image. A brief description of the layers used in ConvNet is:

Input Layer

- It has the raw pixel values of the image as $(w \times h \times c)$ where w and h are the width and height of the image and c is the number of colour channels. In our case, it is $(48 \times 48 \times 1)$ where 1 is for the gray scale images.
- Since the dimensions are fixed, pre-processing needs to be done before feeding the pixels in the input layer.

Convolutional Layer

- This layer computes the dot product between the weights and a small region to which the neurons are connected to in the input layer. The number of filters is passed as one of the hyper parameters which are unique with randomly generated weights. The filter also called a kernel, is convolved with image (i.e. element wise multiplications between filter values and the input pixel values).
- This generates a feature map that acts as feature identifiers sensitive to the edges and the orientations that represent how the pixel values are enhanced. This results in $(w \times h \times f)$, where f is the number of filters used.
- Convolutional layers are followed by a pooling layer that down samples the dimensions along the width and the height to reduce the computational time due to a large number of convolutional layers. MaxPooling is used that reduces the dimensions of the map by a factor of window size and only the maximum pixel value in the original feature map window is retained.

Dense Layer (Fully Connected Layer)

- This layer is fully connected with the output of the previous layer. These are typically used in the last stages of the CNN to connect to the output layer and construct the desired number of outputs. It transforms the features through layers connected with trainable weights.

- This layer identifies the sophisticated features in the image that brings out the entire image.
- Sometimes it becomes prone to over fitting. This is reduced by adding a dropout layer that randomly selects a portion (usually less than 50%) of nodes to set their weights to zero during training.

Output Layer

- This layer is connected to the previous fully connected layer and outputs the required classes or their probabilities.
- Since, in human some emotions are generally an amalgamation of emotions which is computed by probability of each emotion. This is achieved by using softmax layer in the network.

7 Various Attempted Networks, Their Comparisons and the Selection of Proposed Network

The following models have been tried and cross validation is done for the model selection. The results of cross validation are given in Table 1. A pool size of (2, 2), kernel size of (3, 3) and (5, 5) are used. L2 regularization, dropout of 0.3, batch normalization and ‘Uniform Kernel Initializer’ has been used for more accurate results. The models are trained for 60 epochs.

(A) The first architecture we used is inspired from Lenet architecture by Yann LeCun [13]. Lenet is a small network consisting of 2 convolutional layers followed by a dense layer. We modified it by adding an extra dense layer to it. Thus, the network comprises of 2 convolutional layers followed by a MaxPooling layer. 2 dense layers with number of filters 200 and 100 follow. Last layer is a softmax layer that gives the probability of different classes. The hyperparameters are tested and chosen such that the performance metrics are maximised.

(B) The above network is modified by replicating the convolutional layers and the MaxPooling layers to identify the finer edges and patterns more specifically. Thus, this network consists of 2 convolutional layers followed by a MaxPooling layer which is followed by the similar pattern of 2 convolutional layers and MaxPooling layer twice. The dense layers of 64 and 32 filters are subsequently added followed by the final softmax layer. The number of filters in dense layers is reduced to compensate the increased computation time due to the addition of convolutional layers.

(C) Convolutional layers are now added into the second network while the dense layer is kept as before. This is done to check the performance of the addition of the convolutional layers in our model. Hence, this network consists of three blocks of 3 convolutional layers and a MaxPooling layer followed by the 2-dense layer and the final softmax layer. Here, we see that the addition of convolutional layers has a positive impact on the performance metric.

(D) To study the impact of dense layers, we made a network consisting of convolutional layer followed by a MaxPooling layer which is again followed by a

convolutional and a MaxPooling layer followed by a convolutional layer. Dense layer with 3072 filters is then added followed by the output layer (softmax). We observe that increasing the number of filters in the dense layer has a positive influence on the accuracy.

(E) To improve the performance of the previous architecture, we added convolutional layers. The network consists of 2 blocks of 2 convolutional layers and a MaxPooling Layer followed by a convolutional layer and a dense layer with 3072 filters. This is our final model. There is still a scope for improvement. More combinations can be tried, and a proper grid search can be performed with different parameters, which have great computational overhead, but will give much better recognitions.

The emotion recognition accuracy using SVM and all the attempted CNN networks (i.e. A, B, C, D and E) have been depicted in Table 1.

Table 1. Attempted network accuracy

Network	SVM	A	B	C	D	E
Accuracy (%)	46.74	58	50	58	62.32	63.03

8 Proposed Network Architecture

Amongst all attempted CNN networks, we have got maximum accuracy with Network E (Accuracy in Table 1). The algorithm steps are described below. The architecture of Network E has been shown in Fig. 7. The proposed architecture and its working approach to determine the emotion from a facial gesture involves following steps:

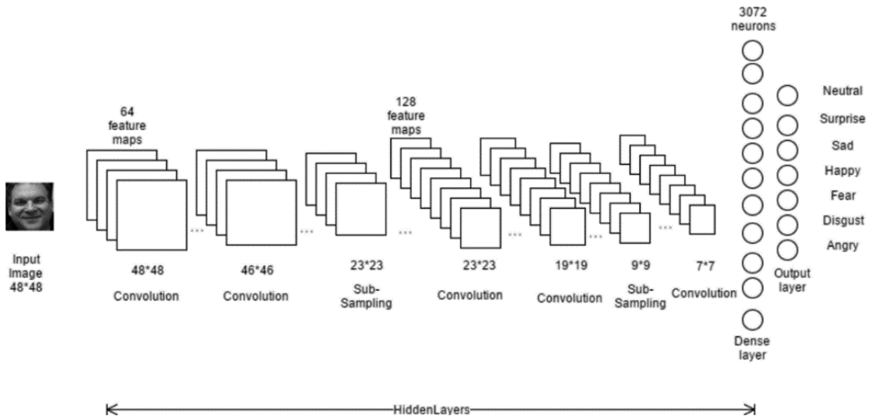


Fig. 7. Proposed network architecture (i.e. Network ‘E’)

 Algorithm 1. Emotion Recognition using Facial Gestures

Input: An Image

Output: Emotion of the faces in image.






Prerequisites: A large dataset with different facial emotions.

1. The dataset is split into training and test sets (70%-30%).
 2. Viola Jones algorithm is applied on all the images in training and test sets for pre-processing.
 3. A simple model is prepared based on some existing model.
 4. Model is then trained on the training set and tested for accuracy on test set.
 5. Grid search is applied for hyper parameter tuning.
 6. Dense and convolution layers are added, and the model is tested for performance metrics.
 7. In case of over fitting, cross validation is done.
 8. Several models are tried with appropriate addition of convolution and dense layers, for better performance.
 9. The best model is then taken as the final model.
-

9 Results

Our proposed model yielded a promising accuracy of 63.03% which is considerably good with less training data. The results for 5 test images are shown in Table 2. The two topmost emotion percentages are highlighted in bold.

Table 2. Emotions results in percentage

Test Image	1	2	3	4	5
Emotions					
Anger	0.5	3.4	2.3	5.75	2.6
Disgust	2.3	2.6	1.2	31.95	2.2
Fear	0.2	9.4	7.1	4.93	8.84
Happy	93.9	0.9	2.3	1.3	33.48
Sad	0.7	35.7	14.3	10.1	9.45
Surprise	1.7	3.2	6.1	2.12	4.8
Neutral	0.45	44.5	66.5	4.63	38.56

We see that our model has correctly identified the emotions in Test Images 1, 3 and 4 in the above table as the highest emotion percent (highlighted in bold) is the true emotion of the face. The accuracy obtained in this way by considering the highest emotion percent as the result in every face and then comparing it with its true emotion

was 63.03%. It was also observed that incorporating the second-best emotion, we can achieve emotions with an accuracy of 67%. For Test Images 2 and 5, we see that the second highest emotion percent (highlighted in bold) is the true emotion of the test face rather than the highest emotion percent. This is analogous to how a human perceives an emotion. It is sometimes difficult to decipher the emotion from a face. As in test image 2, the child appears to be sad to some and neutral to others. While in test image 5, we confuse between happy or neutral.

10 Conclusion

Emotion recognition is still a difficult and a complex problem in computer science because every expression is a mix of emotions. This work tries to address the problem of emotion recognition with deep learning approach using convolutional neural network. First, we have implemented SVM and then attempted 5 different CNN networks (namely A, B, C, D and E) and tested the accuracy. Network E gives the maximum accuracy among all including SVM too. Training and testing of these networks have been performed on FER-2013 database. The system is independent of factors like gender, age, ethnic group, beard, backgrounds and birthmarks. The proposed system is very promising and provides better accuracy in emotion recognition than SVM. The proposed architecture and the algorithm here yield noticeable results. Hence it can motivate the researchers to design the better ‘Deep Learning CNN Architecture’ to enhance the emotion recognition system.

References

1. Carton, J.S., Kessler, E.A., Pape, C.L.: Nonverbal decoding skills and relationship well-being in adults. *J. Nonverbal Behav.* **23**(1), 91–100 (1999)
2. Izard, C.E.: *Human Emotions*. Springer, New York (2013)
3. Happy, S.L., George, A., Routray, A.: A real time facial expression classification system using Local Binary Patterns. In: *Intelligent Human Computer Interaction (IHCI)*, 4th International Conference, pp. 1–5. IEEE (2012)
4. Lee, C.C., Mower, E., Busso, C., Lee, S., Narayanan, S.: Emotion recognition using a hierarchical binary decision tree approach. *Speech Commun.* **53**(9), 1162–1171 (2011)
5. Lopes, A.T., de Aguiar, E., De Souza, A.F., Oliveira-Santos, T.: Facial expression recognition with Convolutional Neural Networks: coping with few data and the training sample order. *Pattern Recogn.* **61**, 610–628 (2017)
6. Hu, T., De Silva, L.C., Sengupta, K.: A hybrid approach of NN and HMM for facial emotion classification. *Pattern Recogn. Lett.* **23**(11), 1303–1310 (2002)
7. Sebe, N., Cohen, I., Gevers, T., Huang, T.S.: Emotion recognition based on joint visual and audio cues. In: *18th International Conference on Pattern Recognition, ICPR*, vol. 1, pp. 1136–1139. IEEE, August 2006
8. Liu, M., Wang, R., Li, S., Shan, S., Huang, Z., Chen, X.: Combining multiple kernel methods on riemannian manifold for emotion recognition in the wild. In: *Proceedings of the 16th ACM International Conference on Multimodal Interaction*, pp. 494–501 (2014)

9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012)
10. FERC 2013: Form 714 – Annual Electric Balancing Authority Area and Planning Area Report (Part 3 Schedule 2) 2006–2012 Form 714 Database, Federal Energy Regulatory Commission (2013)
11. Viola, P., Jones, M.J.: Robust real-time face detection. *Int. J. Comput. Vis.* **57**(2), 137–154 (2004)
12. Weston, J., Watkins, C.: Multi-class support vector machines. Technical report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London (1998)
13. LeCun, Y.: LeNet-5, Convolutional Neural Networks (2015). <http://yann.lecun.Com/exdb/lenet>

Supervised Approaches to Assign Cooperative Patent Classification (CPC) Codes to Patents

Tung Tran¹ and Ramakanth Kavuluru^{1,2}(✉)

¹ Department of Computer Science, University of Kentucky, Lexington, KY, USA
{tung.tran,ramakanth.kavuluru}@uky.edu

² Division of Biomedical Informatics, Department of Internal Medicine,
University of Kentucky, Lexington, KY, USA

Abstract. This paper re-introduces the problem of patent classification with respect to the new Cooperative Patent Classification (CPC) system. CPC has replaced the U.S. Patent Classification (USPC) coding system as the official patent classification system in 2013. We frame patent classification as a multi-label text classification problem in which the prediction for a test document is a set of labels and success is measured based on the micro-F1 measure. We propose a supervised classification system that exploits the hierarchical taxonomy of CPC as well as the citation records of a test patent; we also propose various label ranking and cut-off (calibration) methods as part of the system pipeline. To evaluate the system, we conducted experiments on U.S. patents released in 2010 and 2011 for over 600 labels that correspond to the “subclasses” at the third level in the CPC hierarchy. The best variant of our model achieves $\approx 70\%$ in micro-F1 score and the results are statistically significant. To the best of our knowledge, this is the first effort to reinitiate the automated patent classification task under the new CPC coding scheme.

1 Introduction

A patent can be briefly summarized as a contract between an inventor and a government entity that prevents others from using or profiting from an invention for a fixed period of time; in return, the inventor discloses the invention to the public for the common good. Patents are complex technical and legal documents that exhaustively detail the ideas and scopes of corresponding inventions. The United States Patent and Trademark Office (USPTO) is the government agency responsible for issuing and maintaining patents in the United States. The first U.S. patent was issued in 1790 [26] and since then the USPTO has issued over 9 million patents¹. Hundreds of thousands of patent applications are submitted to the USPTO each year and data suggests that this number will continue to rise heavily in the future. In 2015, there were 589,410 utility patent applications compared to 490,226 in 2010 and 390,733 in 2005 [22].

¹ See Patent No. 9,226,437, granted Jan. 5, 2016. Available at: <http://pdfpiw.uspto.gov/.piw?Docid=09226437>.

A patent application undergoing the review process must meet the novelty criteria in order for it to be published; i.e., it must be original with respect to past inventions. Hence, a *patentability search* is usually performed by inventors, lawyers, and patent examiners to determine whether the patent constitutes a novel invention. The manual process can be a strain on time and resource because of the scientific expertise needed to verify patentability. A patent classification system (PCS) is vital for organizing and maintaining the vast collection of patents for later lookup and retrieval. We first provide a brief dissection of a typical patent document before describing the various PCSs and formalizing the problem of assigning PCS codes to a patent. Since we are primarily concerned with patents in a U.S. context, mentions of patents in the remainder of this paper implicitly refer to U.S. patents unless otherwise stated.

A patent consists of several sections including title, abstract, description, and claims. From our observation, the title and abstract are consistent in length with those of a typical research paper, while description is much larger in detail and scope. The claims section is unique in that it describes the invention in units of “innovation”; each claim corresponds to a novel aspect of the invention and is conveyed in nuanced legal terminology. According to Tong et al. [20], the claims of a patent can actually be considered a collection of separate inventions; together they can be used to determine the true measure of a patent. A patent document additionally contains structured bibliographical information such as inventors, lawyers, publication date, application date, application number, and technology class assignments. Technology class assignments are available for each of the three PCSs: the U.S. Patent Classification (USPC) system, the Cooperative Patent Classification (CPC) system, and the International Patent Classification (IPC) system. The USPC has been the official PCS used and maintained by the USPTO since the first patent was issued. In January 1, 2013, however, it was replaced by the CPC system as a joint effort by the USPTO and European Patent Office (EPO) to promote patent document compatibility at the international level [24]. The CPC is intended to be a more detailed extension of the International Patent Classification (IPC) system². As part of the transition, all US patent documents as of 1836 have been retroactively annotated with CPC codes using “an electronic concordance system” [18].

In CPC, the classification terms/labels (CPC codes) are organized in a taxonomy – a tree with each child label being a more specific classification of its parent label; that is, there is an *IS-A* relation between a label and its parent. A single patent can be manually assigned one or more labels corresponding to the leaf nodes (of the taxonomy) by a patent examiner. There are five levels of classification: *section*, *class*, *subclass*, *group*, and *subgroup*. As of January 2015, there are 9 sections, 127 classes, 654 subclasses, 10633 groups, 254794 subgroups in the CPC schema [23]. An example of a leaf label and its parent labels can be

² According to the Guide to the CPC [24], “unless stated otherwise, the rules and principles are identical” to that of IPC, which can be found in Guide to the IPC [27].

Table 1. Overview and example of CPC hierarchical taxonomy. The label count at each level is computed from the January 2015 version of the CPC scheme [23].

Level	Count	Example Label	Example Label Description
Section	9	H	Electricity
Class	127	H01	BASIC ELECTRIC ELEMENTS
Subclass	654	H01C	RESISTORS
Group	10633	H01C 3	Non-adjustable metal resistors made of wire or ribbon, e.g. coiled, woven or formed as grids
Subgroup	254794	H01C 3/08	Dimension or characteristic of resistive element changing gradually or in discrete steps from one terminal to another

observed in Table 1. Since CPC is an extension of IPC, many of the characteristics of CPC as described can be similarly observed in IPC.

In this paper, we propose a supervised machine learning system for the classification of patents according to the newly implemented CPC system. To our knowledge, ours is the first automatic patent classification effort for CPC based on literature review. Our system exploits the hierarchical nature of the CPC taxonomy as well as the citation records³. CPC codes appear in order of how adequately a code represents the invention [27]; here, we neglect the ordering and treat the problem as a multi-label classification problem. That is, our prediction is a set of labels, a task that is more challenging and comprehensive compared to past work that focuses on predicting a single “main” IPC label for each test patent. This is because many inventions span across multiple technological domains and this level of nuance cannot be captured in a single CPC code. As in past work that deal with the older IPC system, we collapse all CPC labels of a patent to their subclass representations and make predictions at the *subclass* level (row 3 of Table 1). We use real-world patent documents published by the USPTO in the years 2010 and 2011 to train, tune, and evaluate our proposed models. Moreover, we publicly release the dataset⁴ used in training and evaluating our system to stimulate further research in the area.

2 Related Work and Background

Fall et al. [4] explored the task of patent classification for IPC using various supervised algorithms such as support vector machines (SVM), naïve Bayes, and

³ These are other older patents that appear in the *References Cited* section and are sometimes mentioned within the description of a new patent being considered for automated coding.

⁴ <http://patents.ttran.net>.

k -nearest neighbors (k -NN). Their experiments used the title, first 300 words, and claims as the predictive scope for feature extraction. The system proposed does not attempt to make predictions on the correct set of labels but rather produces an exhaustive label ranking for each patent on which custom “precision” metrics are used to evaluate performance. For instance, the prediction for a test document is deemed correct if the top-ranked predicted label matches the first label of the patent. The authors also conducted experiments with variants of this metric such as whether any of the top-3 prediction labels matches the first label of the patent or whether the top-ranked prediction appears anywhere in the ground truth list of IPC labels. Their study concluded that SVM was superior to other methods under similar conditions.

Liu and Shih [11] proposed a hybrid system for USPC classification using patent network analysis in addition to traditional content-based features. Their approach is based on first constructing a network graph with patents, technology classes, inventors, and assignees as nodes. Edges indicate connectivity and edge weights are computed using a custom relation metric. A prediction for a test patent is made by looking at its nearest neighbors based on a relevance measure as determined by the constructed patent network. Li et al. [10] exploit patent citation records by proposing a graph kernel that captures informative features of the network. Richter and MacFarlane [14] showed that in some cases using features based on bibliographical meta-data can improve classification performance. Other studies [2, 8] found that exploiting the semantic structure of a patent such as its individual claims can result in similar gains. Automatic patent classification in the literature has primarily focused on IPC [2–4] or USPC [10, 11], and we observe k -NN to be a popular approach for many proposed systems [8, 11, 14]. When targeting IPC, classification is typically performed at the class or subclass level. This choice is motivated by the fact that labels at the subclass level are fairly static moving forward while group and subgroup labels are more likely to undergo revision with each update [4] of PCS system.

3 Datasets

The dataset used in our experiments consist of utility patent documents published by the USPTO in 2010 and 2011, not including pending patent applications. For the 2010 and 2011 datasets, there are 215,787 and 221,206 patent documents respectively. Specifics about training and test set splits for supervised experiments are outlined in Sect. 5. The patents documents are freely available in HTML format at the USPTO website, although not in a readily machine-processable format. As outlined earlier, each patent contains the text fields *title*, *abstract*, *description*, and *claims*. Furthermore, each document is annotated with a set of one or more CPC labels. From the dataset, we counted 613 unique CPC subclasses which correspond to the range of candidate labels for this predictive task.

The distribution of CPC subclasses is skewed with some codes such as *G06F* (Electrical Data Processing), *H01L* (Semiconductor Devices), and *H04L* (Transmission of Digital Information) dominating the patent space at 5.9%, 4.6%, and

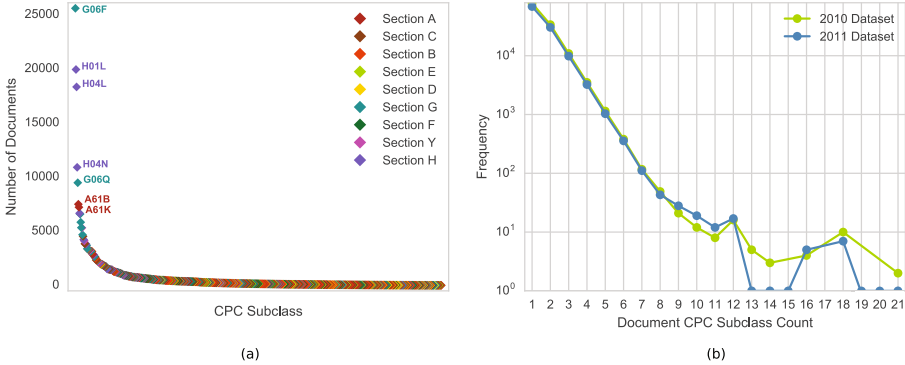


Fig. 1. Overview of CPC subclass level statistics within the 2010 and 2011 datasets for (a) CPC subclasses along the x -axis in order of label frequency—only those above the 99th percentile are labeled—and (b) frequency of the number of CPC subclasses that appear in a document.

4.1% document assignments respectively as seen in Fig. 1(a). The distribution of the number of CPC subclass assignments per document is likewise skewed such that the average number of labels per patent is only 1.76. Indeed, approximately 60% of documents contain only a single subclass while some outlier documents may have as many as 21 subclasses as shown in Fig. 1(b).

4 Methods: Label Scoring, Reranking, and Thresholding

As indicated earlier, the CPC taxonomy is hierarchical and takes upon a tree structure so that each label exists as a non-root node in the tree; henceforth, we refer to nodes and labels interchangeably. Since we are concerned with classification at the *subclass* (or third) level in the hierarchy, nodes at the *subclass* level are considered leaf-nodes in our experiments⁵. However, in practice it is possible to choose any target depth d in the hierarchy as the level at which labels are trained on (and predictions are made) essentially assuming d is the leaf-node level. The following notation is used in the formulation of our methodology. Let L^i be the set of all labels at level i in the CPC tree. For a leaf node $c \in L^d$, we define $[c]_i$ as the ancestor node of c at level i such that $[c]_i \in L^i$ and $i \leq d$. Given the tree structure, a node has unique ancestors at each level above it. As a special case, we have $c = [c]_i$ if $i = d$.

⁵ Although code assignment in reality is only done at the subgroup level, we can collapse such deeper codes to the subclass level by simply removing code components specific to the deeper nodes. In Table 1, code for resistors (row 3) can be obtained from the more specific subgroup code (row 5) by removing the “3/08” part. Typically, due to extremely high sparsity, code assignments are carried out at a higher level in preliminary studies including our current attempt and other automated PCS efforts.

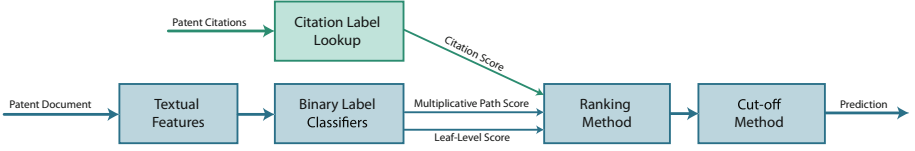


Fig. 2. The pipeline of the proposed framework. The core framework produces scores for each patent while the ranking and cut-off components jointly make a label set prediction based on these scores.

Basic multi-label pipeline. Figure 2 represents the high level skeleton of our approach, a pipeline setup that is quite common in multi-label classification scenarios [28] where binary classifiers, one per label, are used first based on n -gram features. Besides scores from these classifiers, additional domain-specific features that are not directly related to the lexical n -gram features of the text are also used to score the labels. Subsequently, the scores are used to rank all labels for a given input instance. Given the sparsity concerns, the default threshold of the binary classifiers might not be suitable for infrequent labels. So instead a so called cut-off or calibration method [5, Sect. 4.4] is used to make a partition of all labels where the top few labels that form one of the partitions are considered as the final predictions. This approach has been used in obtaining medical subject headings for biomedical research articles [12, 15] and assigning diagnosis codes to electronic medical records [7]. We will elaborate on each of the components in Fig. 2 in the rest of this section with a focus on novel CPC and patent specific variations we introduce in this paper. But first, we outline the configuration for the binary label specific classifiers.

Lexical features for binary classifiers. For the textual features, we extract unigram and bigram features from the title, abstract, and description fields. We found that including textual features from the claims field tend to have a neutral or negative impact on performance while adding drastically to the feature space. We suspect that unigram and bigram features are unable to adequately capture the nuance in legal language and style as presented in the itemized claims section. To further reduce the feature space, we apply a lowercase transformation prior to tokenization, remove 320 popular English stop words, and ignore terms that occur in fewer than ten instances in the entire corpus. We apply the well-known *tf-idf* transformation [16] to word counts to produce the final document-feature matrix. In total, there are nearly 6.8 million n -gram features in the final document-feature matrix.

Training label-specific classifiers. Our system uses a top-down approach [17] such that a local binary classifier is trained for each label in the taxonomy (including the non-leaf nodes). Let f_c be the classifier associated with the label $c \in L^1 \cup \dots \cup L^d$. The classifier f_c is trained on a set of positive and negative examples, such that the set of positive example set $[M_c]^+$ includes only documents that are labeled with c and the set of negative examples $[M_c]^-$ is the set of patents that are not

labeled with c . Since there typically many more negative examples than positive examples for a given label, we use only a random subsample of negative examples such that $||M_c^+| = ||M_c^-|$. This under-sampling of the majority class is a well-known idea that seems to fare well in imbalanced datasets [25]. A simple logistic regression (LR) model is used for each classifier which expresses the output in the form of a probability estimate in the range of $[0, 1]$.

4.1 Label Scoring

Since the classification task is multi-label, the prediction for some patent x is necessarily a subset of all possible labels at level d in the CPC hierarchy. A natural approach is to rank the labels based on some *scoring* function with respect to x , then truncate the ranked list after some cut-off k such that only the top k labels appear in the predicted set. We define three such scoring functions to be used as the basis for our label ranking methods: leaf-level score, hierarchical multiplicative-path score, and citation score. Let $L^i(x)$ be the set of labels at level i assigned to patent x . Each scoring function takes two parameters: input patent x and a leaf-node label $c \in L^d$.

The leaf-level score

$$S_L(c, x) = f_c(x) \tag{1}$$

is simply the probability $f_c(x)$ output by the binary classifier for c at level d in the hierarchy. Since the problem is *mandatory leaf-node prediction*, using this score alone in label ranking is referred to as *flat classification* in the literature [17]. The multiplicative-path score is the geometric mean⁶ of probability outputs for each classifier along the path from the leaf-node label to the root, or more formally,

$$S_M(c, x) = \left(\prod_{i=1}^d f_{[c]_i}(x) \right)^{1/d} . \tag{2}$$

Both $S_M(c, x)$ and $S_L(c, x)$ produce a real number in $[0, 1]$. Next, we define the citation score that is specific to the patent domain. Let $R(x)$ be the set of patents cited by x . For a given patent x and a candidate label c , the raw citation score

$$S_R(c, x) = |\{t : t \in R(x) \wedge c \in L^d(t)\}| , \tag{3}$$

which intuitively counts the number of cited patents that also are assigned our candidate code c . This is not unlike the k -NN approach where similar instances' code sets are exploited to score candidate codes for an input instance. However, in this work, we simply used the cited patents as neighbors drastically reducing the notorious test time inefficiency issues for nearest neighbor models. The citation

⁶ The product of d probabilities is significantly reduced in magnitude especially for large values of d . Here, use of the geometric mean is intended to restore the magnitude of the resulting product while ensuring that it remains in $[0, 1]$. This eases the interpretability of intermediate probability outputs while guarding against practical concerns such as floating point underflow.

score above is simply a count but we linearly rescale it across all leaf-level labels such we obtain a value in $[0, 1]$ range via

$$S'_R(c, x) = \frac{S_R(c, x) - \min Z(x)}{\max Z(x) - \min Z(x)} \quad (4)$$

where $Z(x) = \{S_R(c, x) : c \in L^d\}$.

4.2 Label (re)ranking

The three different scoring functions in Eqs. (1), (2), and (4) offer evidences of the relevance of a label c for a test instance x . At this point, we do not know how much each of these contributes to an effective ranking model. Thus to leverage them, we combine them into a final score – for the purpose of label ranking – using two approaches

1. The first approach involves a *grid search* [6] over weights $p_1 \geq 0$ and $p_2 \geq 0$, where $0 \leq p_1 + p_2 \leq 1$, for the combined scoring function

$$S(c, x) = p_1 \cdot S_L(c, x) + p_2 \cdot S_M(c, x) + (1 - p_1 - p_2) \cdot S'_R(c, x), \quad (5)$$

that maximizes the micro F1-score over a validation dataset. Note that the coefficients of the three constituent functions in Eq. (5) add up to 1. Hence, given each score is also in $[0, 1]$, this ensures $S(c, x) \in [0, 1]$.

2. The second approach uses what is known in the literature as *stacking* [9], a popular meta-learning trick. Here, we train a new binary LR classifier for each label at level d using scores S_L , S_M , and S'_R computed from each patent example in the validation set as features.

4.3 Label Cut-Off/Thresholding

As mentioned earlier, once we have a final ranking among leaf labels for some patent x , we need to choose the optimal cut-off k tailored specifically to x . Thus only the top k labels are included in the final prediction. One simple approach is to train a linear regression model to predict the number of labels based on core n -gram features [12] and a few select domain-specific features. We found that adding length-based features such as character-count and word-count of text fields resulted in poor performance. Instead, the following features were selected given a patent x : the number of claims made by x , the number of inventors associated with x , and descriptive statistics based on the distribution of known label counts associated with the set of citations $R(x)$. We refer to this approach as the *linear* cut-off method.

The linear cut-off approach will serve as a reasonable baseline. However, the skewed distribution of label counts presents a source of difficulty. Approximately 60.95% and 26.39% of patents in the 2010 dataset have one and two labels respectively. The remaining 12.66% of patents are exceptions in that they have more than two labels with some patents having as many as 21 labels. A simple

linear regression approach has the tendency to overestimate the label count, as such we consider a more sophisticated version of this method using a two-level top-down learning model. In the first level, we train a binary classifier to predict whether, for some test patent, the case is common (one or two labels) or an exception (three or more labels). If it is predicted common, we train another binary classifier at the second layer to predict whether the number of labels is one or two. If it is predicted to be the exception case, we instead use a linear regression model to predict a real number value, which is rounded to the nearest natural number and used as the outcome for the label-count predictor. We refer to this approach as the *tree* cut-off method and is inspired by the so called “hurdle” process in regression methods for count data [1].

Finally, we propose a more involved cut-off method that does not rely on supervised learning. In this alternative method, referred to as *selective cut-off*, there are two steps. (1) Recall our final scoring function $S(c, x)$ from Eq. (5) used to rank labels before applying cut-off. Suppose the label ranking is $c_{i_1}, \dots, c_{i_{|L^d|}}$, $i_j \in \{1, \dots, |L^d|\}$, we choose the cut-off k based on

$$\operatorname{argmax}_{k \in \{1, \dots, |L^d| - 1\}} S(c_{i_k}, x) - S(c_{i_{k+1}}, x).$$

This has the effect of choosing the cut-off at the greatest “drop-off point” in the ranking with respect to the score. (2) Performing the first step alone is prone to overestimating the actual k , so additional pruning is required. Suppose the top k labels are c_{i_1}, \dots, c_{i_k} . As a second step, we further remove a label from this ranking if its rank is higher than the average label count over all patents (in training data) to which it is assigned. Let A_j be the mean label count over all patents in the training set with label c_{i_j} for some $1 \leq j \leq k$. We remove c_{i_j} from the final list if $j > A_j$. Intuitively, for example, a label that typically appears in label sets of size 3 on average is not likely to rank 4th or higher when generalizing to unseen examples. In a way, this leverages the thematic aspects of patents and how certain themes typically lend themselves to a narrow or broad set of patent codes.

5 Experiment and Results

For our experiments, we optimized hyperparameters (from Eq. (5)) on the 2010 dataset using a split of 70% and 30% for training and development, respectively. We use this 2010 dataset exclusively for tuning hyperparameters. We evaluate the variants of our system by training and testing on the 2011 dataset (with the 70%–30% split) using the hyperparameters optimized on the 2010 dataset to simulate the prediction process – learn on existing data to predict for future patents. Since we are framing it as a multi-label problem with class imbalance concerns, we measure classification performance based on the popular micro-averaged F1 [21] measure. We evaluate all combinations of the proposed ranking and cut-off methods on top of the core framework (as described in Sect. 4). For each variant of our system, we perform 30 experiments each with a

random train-test split of the 2011 dataset and record the micro-averaged F1/Precision/Recall as shown in Table 2. The macro-averaged F1 measure that gives equal importance to all labels regardless of their frequency is additionally recorded in Table 3.

Table 2. Results comparing variants of our method in micro-averaged F1

	Micro-P (%)	Micro-R (%)	Micro-F (%)
Grid Ranking + Linear Cut-off	69.951 ± 0.072	68.392 ± 0.051	69.146 ± 0.034
Grid Ranking + Tree Cut-off	69.926 ± 0.059	68.400 ± 0.046	69.154 ± 0.037
Grid Ranking + Selective Cut-off	73.749 ± 0.057	66.418 ± 0.040	69.892 ± 0.033
Stacked Ranking + Linear Cut-off	63.391 ± 0.236	47.804 ± 0.175	54.505 ± 0.200
Stacked Ranking + Tree Cut-off	62.517 ± 0.230	48.799 ± 0.190	54.812 ± 0.206
Stacked Ranking + Selective Cut-off	63.423 ± 0.238	47.52 ± 0.177	54.505 ± 0.200

Table 3. Results comparing variants of our method in macro-averaged F1

	Macro-F (%)
Grid Ranking + Linear Cut-off	74.281 ± 0.031
Grid Ranking + Tree Cut-off	74.274 ± 0.031
Grid Ranking + Selective Cut-off	74.262 ± 0.032
Stacked Ranking + Linear Cut-off	57.452 ± 0.214
Stacked Ranking + Tree Cut-off	57.652 ± 0.214
Stacked Ranking + Selective Cut-off	57.403 ± 0.214

Among the proposed approaches, the *grid* based ranking and *selective* cut-off combination achieved a micro-F1 of almost 70%, which is best performance (row 3 of Table 2). Based on non-overlapping confidence intervals, we also see the differences in performance with respect to other variants is statistically significant for this particular combination. From Table 2, it is clear that the *stacked* ranking approach performs poorly compared to the *grid* ranking approach and is observed to be approximately 15 points worse in micro-F1 due to the low recall. Among variants that use the *grid* ranking method, the *selective* cut-off approach is able to achieve superior precision with only a minor dip in recall. It can also be observed that, in terms of macro-F1, the *grid* ranking approach is still superior by far; with *grid* ranking, *linear* cut-off has a higher average macro-F1 over

other cut-off methods but the gains are not statistically significant. Moreover, we note that the macro-F1 score is greater than the micro-F1 score for each respective method combination, which may suggest that the system is generally performing better on the low-frequency labels and worse on the popular ones [19]. This is a counterintuitive outcome and needs further examination as part of our future work. However, it could also be due to the case that some infrequent labels are very specific to certain esoteric domains for which the language used in the patents is highly specific.

6 Conclusion

In this paper, we proposed an automated framework for classification of patents under the newly implemented CPC system. Our system exploits the CPC taxonomy and citation records in addition to textual content. We have evaluated and compared variants of the proposed system on patents published in years 2010 and 2011. As a take away, we demonstrated that using the proposed framework with *grid* ranking (based on three different scoring functions) and the *selective* cut-off method outperform other variants of the system. In this work, we used logistic regression as the base classifier since it is fast and uses relatively fewer parameters. In future work, we propose to combine the information in the CPC hierarchy as a component of recurrent and convolutional neural networks for multi-label text classification using the cross-entropy loss [13].

Acknowledgements. We thank anonymous reviewers for their honest and constructive comments that helped improve our paper’s presentation. Our work is primarily supported by the National Library of Medicine through grant R21LM012274. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

References

1. Cameron, A.C., Trivedi, P.K.: Regression Analysis of Count Data, vol. 53. Cambridge University Press, Cambridge (2013)
2. Don, S., Min, D.: Feature selection for automatic categorization of patent documents. *Indian J. Sci. Technol.* **9**(37), 1–17 (2016). Kindly check and confirm the edit made in Ref. [2]
3. Eisinger, D., Tsatsaronis, G., Bundschuh, M., Wieneke, U., Schroeder, M.: Automated patent categorization and guided patent search using IPC as inspired by MeSH and PubMed. *J. Biomed. Semant.* **4**(S1), 1–23 (2013)
4. Fall, C.J., Töröcsvári, A., Benzineb, K., Karetka, G.: Automated categorization in the international patent classification. *ACM SIGIR Forum* **37**(1), 10–25 (2003)
5. Gibaja, E., Ventura, S.: A tutorial on multilabel learning. *ACM Comput. Surv. (CSUR)* **47**(3), 52 (2015)
6. Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al.: A practical guide to support vector classification (2003). <https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>

7. Kavuluru, R., Rios, A., Lu, Y.: An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artif. Intell. Med.* **65**(2), 155–166 (2015)
8. Kim, J.-H., Choi, K.-S.: Patent document categorization based on semantic structural information. *Inf. Process. Manag.* **43**(5), 1200–1215 (2007)
9. Kotsiantis, S.B., Zaharakis, I., Pintelas, P.: Supervised machine learning: a review of classification techniques (2007)
10. Li, X., Chen, H., Zhang, Z., Li, J.: Automatic patent classification using citation network information: an experimental study in nanotechnology. In: *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries*, pp. 419–427. ACM (2007)
11. Liu, D.-R., Shih, M.-J.: Hybrid-patent classification based on patent-network analysis. *J. Assoc. Inf. Sci. Technol.* **62**(2), 246–256 (2011)
12. Liu, K., Peng, S., Wu, J., Zhai, C., Mamitsuka, H., Zhu, S.: MeSHLabeler: improving the accuracy of large-scale MeSH indexing by integrating diverse evidence. *Bioinformatics* **31**(12), i339–i347 (2015)
13. Nam, J., Kim, J., Loza Mencía, E., Gurevych, I., Fürnkranz, J.: Large-scale multi-label text classification — revisiting neural networks. In: *Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014. LNCS (LNAI)*, vol. 8725, pp. 437–452. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44851-9_28
14. Richter, G., MacFarlane, A.: The impact of metadata on the accuracy of automated patent classification. *World Pat. Inf.* **27**(1), 13–26 (2005)
15. Rios, A., Kavuluru, R.: Analyzing the moving parts of a large-scale multi-label text classification pipeline: experiences in indexing biomedical articles. In: *2015 International Conference on Healthcare Informatics (ICHI)*, pp. 1–7. IEEE (2015)
16. Salton, G., McGill, M.J.: *Introduction to Modern Information Retrieval*. McGraw-Hill, New York (1986)
17. Silla, C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. *Data Min. Knowl. Discov.* **22**(1–2), 31–72 (2011)
18. Simmons, H.J.: Categorizing the useful arts: part, present, and future development of patent classification in the United States. *Law Libr. J.* **106**, 563 (2014)
19. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **45**(4), 427–437 (2009)
20. Tong, X., Frame, J.D.: Measuring national technological performance with patent claims data. *Res. Policy* **23**(2), 133–141 (1994)
21. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: *Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook*, pp. 667–685. Springer, Boston (2010). https://doi.org/10.1007/978-0-387-09823-4_34
22. U.S. Patent and Trademark Office: U.S. Patent Statistics Chart. https://www.uspto.gov/web/offices/ac/ido/oeip/taf/us_stat.htm (2016). Accessed 30 Nov 2016
23. U.S. Patent and Trademark Office and European Patent Office: Cooperative Patent Classification Scheme in Bulk. <http://www.cooperativepatentclassification.org/cpcSchemeAndDefinitions/Bulk.html> (2015). Accessed 01 Feb 2015
24. U.S. Patent and Trademark Office and European Patent Office: Guide to the CPC. <http://www.cooperativepatentclassification.org/publications/GuideToTheCPC.pdf> (2015). Accessed 30 Nov 2016
25. Wallace, B.C., Small, K., Brodley, C.E., Trikalinos, T.A.: Class imbalance, redux. In: *2011 IEEE 11th International Conference on Data Mining (ICDM)*, pp. 754–763. IEEE (2011)

26. Wolter, B.: It takes all kinds to make a world-some thoughts on the use of classification in patent searching. *World Pat. Inf.* **34**(1), 8–18 (2012)
27. World Intellectual Property Organization: Guide to the IPC. http://www.wipo.int/export/sites/www/classifications/ipc/en/guide/guide_ipc.pdf (2016). Accessed 30 Nov 2016
28. Zhang, M.-L., Zhou, Z.-H.: A review on multi-label learning algorithms. *IEEE Trans. Knowl. Data Eng.* **26**(8), 1819–1837 (2014)

A Betweenness Centrality Guided Clustering Algorithm and Its Applications to Cancer Diagnosis

R. Jothi(✉)

Department of Computer Engineering, School of Technology,
Pandit Deendayal Petroleum University, Gandhinagar, India
jothi.r@sot.pdpu.ac.in

Abstract. Clustering has become one of the important data analysis techniques for the discovery of cancer disease. Numerous clustering approaches have been proposed in the recent years. However, handling of high-dimensional cancer gene expression datasets remains an open challenge for clustering algorithms. In this paper, we present an improved graph based clustering algorithm by applying edge betweenness criterion on spanning subgraph. We carry out empirical analysis on artificial datasets and five cancer gene expression datasets. Results of the study show that the proposed algorithm can effectively discover the cancerous tissues and it performs better than two recent graph based clustering algorithms in terms of cluster quality as well as modularity index.

Keywords: Clustering · Cancer diagnosis · Betweenness Spanning subgraph · Minimum spanning tree

1 Introduction

With recent advances in microarray technology, the amount of gene expression data handled by the biologists has seen a rapid growth. This has led to an increasing demand for exploratory analysis to identify structural patterns in the data for eg., molecular signatures of cancer cells [14]. Clustering is one of the popular machine learning techniques for the discovery of cancerous tissues (samples) from microarray gene expression data. The goal of clustering is to group the similar set of objects into a cluster so that objects within a cluster are more similar as compared to the objects in other clusters [6].

Numerous clustering methods have been proposed for cancer diagnosis (see [8, 9, 14] and references therein). Hierarchical clustering is one of the earliest methods applied on gene expression datasets to uncover diseases [4]. Similarly, K-means also has gained wide popularity in analyzing cancer gene expression datasets [15]. The major problem with these traditional approaches is that they are sensitive to noise. Moreover, they may not separate the clusters of arbitrary shapes, sizes and densities in high dimensional space [8, 16].

Recently, graph based clustering approaches have shown improvement over traditional clustering methods in cancer diagnosis [9, 12, 17]. These approaches model a gene expression dataset as an undirected graph and try to identify the clusters using graph's edge-structure properties. As gene datasets are highly interconnected [8], these algorithms have the inherent advantage of retrieving clusters through graph connectivity property.

Clustering gene network by exploiting neighborhood properties of each gene in the network is widely studied in the microarray analysis [1, 2, 5, 13]. Nearest Neighbor Network (NNN) algorithm makes use of mutual nearest neighborhood principle to extract more complex and biologically relevant clusters [5]. The NNN algorithm achieves higher precision in detecting functionally related genes. However, accuracy of the algorithm depends on size of the neighborhood [7]. Ruan et al. [13] have proposed a method to construct rank-based co-expression networks that utilize rank-transformed similarities instead of conventional value-based similarities. They have shown that the rank-based co-expression networks can better capture global topology of the network, identifying both strongly and weakly co-expressed modules. Once a rank-based gene network is constructed, a partitioning algorithm with an objective function called modularity is used to automatically determine the optimal partitioning and the number of partitions. Results of the partitioning relies on rank transformation, which in turn depends on the user-specified threshold for rank value.

Dost et al. [3] have proposed a clustering method T-CLUST that exploits coconnectedness to efficiently cluster large, sparse expression data. Basis of this approach is Tanimoto Coefficient (TC), which measures coconnectedness, i.e., whether two vertices are connected to the similar set of vertices or not. Baya et al. [1] have proposed a penalized K-Nearest Neighbor graph based clustering for gene expression analysis (PKNNG). They first construct KNN graph of the given gene expression data for low value of K in the range 3 to 7. If the KNNG is disconnected, they connect the subgraphs by adding edges with an exponentially penalized weight. Experimental results have shown that hierarchical clustering with PKNNG metric has obtained improved results as compared to basic metric. However, finding KNNG of high dimensional datasets is computationally expensive. Also if the dataset has uneven distribution of cluster sizes, larger cluster is broken into multiple subgraphs which also get penalized weight assignment.

More recently, Minimum spanning tree (MST) based graph clustering algorithms have drawn much attention for gene expression analysis. MST-based clustering algorithm using Betweenness-heuristics (B-MST) proposed by Pirim et al. [12] identified more biologically relevant clusters using the betweenness measure, which is defined as the number of times a given edge appears on the shortest path between any pairs of nodes [11]. Eigenanalysis on MST-based neighborhood graph (E-MST) algorithm proposed by Jothi et al. [9] has been shown to provide improved clustering results on gene expression datasets. E-MST employs a spectral multi-partitioning method on neighborhood graph obtained by multiple rounds of MSTs and also presents a relation between algebraic structural properties of the neighborhood graph and the separability of clusters. Both these

methods have employed the neighborhood relation depicted by the MST to determine biologically relevant clusters from gene expression datasets.

Inspired from the results of MST based approaches for gene expression clustering [9, 10, 12], this paper presents an improved clustering algorithm using betweenness centrality. Although betweenness centrality has been widely applied for clustering, most of these methods have assumed that the data is available in the form of a graph. But for cluster analysis of relational data such as gene datasets, construction of suitable graph itself is not a trivial problem. As cluster identification using betweenness measure heavily depends on the graph representation, devising a suitable graph structure for betweenness based clustering algorithms is an essential preprocessing task especially when applied on cancer gene datasets.

Contribution of the proposed algorithm consists in adapting betweenness-centrality on a sparse spanning subgraph so that iterative edge removal takes only few iterations. First, a spanning subgraph of the given dataset is constructed from the union of edge-disjoint MSTs. Such a spanning subgraph effectively represents a set of objects as a collection of dense modules linked with a small number of inter-module edges. Thus identifying and separating such inter-module edges naturally results in a set of intrinsic clusters. So betweenness centrality is employed to locate and remove the inter-module edges in order to get required clusters. We test the proposed algorithm on cancer tissue datasets taken from Broad Cancer Institute. Experimental results show that the proposed algorithm achieves improved identification of cancerous tissues.

The Rest of this paper is organized as follows. The proposed algorithm is described in Sect. 2. Experimental results are reported and discussed in Sect. 3. Conclusion and future scope are given in Sect. 4.

2 Clustering Using Betweenness Centrality on Spanning Subgraph

Let $X = \{x_1, x_2, \dots, x_n\}$ denote the given dataset with n objects (tissues) each having some d attributes. Let k be the number of clusters in X , which may or may not be known in advance. The proposed algorithm has two steps. As a first step, an undirected graph $G = (V, E)$ of the dataset X is obtained, where each $v_i \in V$ corresponds to the object x_i and each edge (v_i, v_j) is the dissimilarity score (e.g., Euclidean distance) of the objects x_i and x_j . A spanning subgraph of the dataset is obtained from the union of edge-disjoint MSTs. Use of edge-disjoint MSTs has been well studied for the problem of clustering [9]. The proposed algorithm obtains a spanning subgraph as follows [9]. The first round of MST $T_1 = (V, E_1)$ is constructed from G . Then, the second round of MST T_2 is constructed from $G = (V, E \setminus E_1)$. Similarly, we can obtain N^{th} round of MST from $G = (V, E \setminus \cup_{j=1}^{N-1} E_j)$. The spanning subgraph SG is formed by combining the edges of T_1, T_2, \dots, T_N . Each node in SG is tightly connected to its nearest neighbors of its own clusters and loosely connected to objects

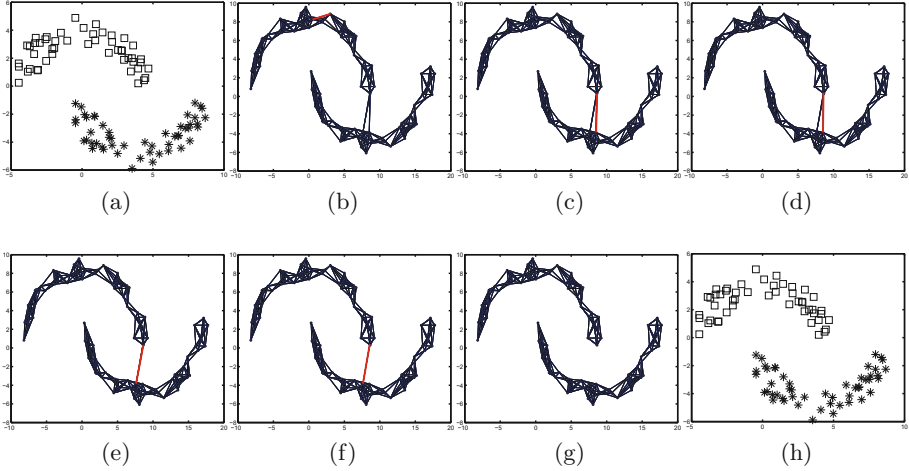


Fig. 1. Cluster separation by removing high-betweenness edges from spanning subgraph. (a) Hal-moon dataset with two clusters represented by upper and lower half spirals. (b, c, d, e, f) Iterative removal of high betweenness edges from the spanning subgraph, high-betweenness edges during each iteration is marked in red color. (g) Dis-connected modules of spanning subgraph after removing high betweenness edges. (h) Resulting clusters from (g). (Color figure online)

of different clusters. Thus the neighborhood information obtained by such a spanning subgraph effectively represent the degree of bridging between clusters.

Consider an example dataset, called as half-moon dataset, and its spanning subgraph shown in Fig. 1a. It is obvious from Fig. 1b that nodes of the same cluster are having more number of adjoining edges, whereas nodes from different clusters are loosely connected. The edges connecting nodes of different clusters are actually the bottleneck edges or bridging edges. Identifying and removing such bridging edges will separate the graph into number of disconnected modules. To achieve this, we apply edge-betweenness centrality.

Edge-betweenness EB of an edge e is computed as sum of the fraction of all-pairs shortest paths that pass through e [11]:

$$EB(e) = \sum_{v_i \in V} \sum_{v_j \in V \setminus v_i} \frac{\sigma_{v_i v_j}(e)}{\sigma_{v_i v_j}}, \quad (1)$$

where $\sigma_{v_i v_j}(e_i)$ is the number of shortest paths between the vertices v_i and v_j that pass through the edge e and $\sigma_{v_i v_j}$ is the total number of shortest paths that run from v_i and v_j . We adapt the shortest-path procedure given in [11] for the calculation of edge-betweenness. We first compute the betweenness of all the edges in SG and remove the edge with high betweenness. Once again betweenness of remaining edges are recomputed and this iterative process continues until the graph SG gets decomposed into k modules or clusters (see Fig. 1). One inherent advantage of applying betweenness centrality on a sparse MST-based spanning

Algorithm 1. *Betweenness-based clustering algorithm on spanning subgraph*

Input: *Dataset X.*

Output: *k clusters.*

1. Construct an undirected graph $G = (V, E)$ from the given dataset, where each v_i in V denotes a cancer tissue sample and each edge $e_l = (v_i, v_j)$ in E denotes dissimilarity (Euclidean distance) between the tissue samples v_i and v_j .
 2. Construct adjacency matrix W of G , where each w_{ij} denotes the weight of edge (v_i, v_j) .
 3. Let N denote number of rounds of MSTs needed and i denote current round index. Also set $i = 1$.
 4. Let SG be the spanning subgraph, where $SG_{ij} = 0$ for $1 \leq i \leq n$ and $1 \leq j \leq n$.
 5. Find i^{th} round MST of the dataset T_i from W and add edges of T_i to SG .
 6. Exclude edges of T_j from W , where $1 \leq j \leq i - 1$.
 7. Repeat steps 4 to 7 until $i \leq N$.
 8. For each edge e_l in SG , find betweenness centrality using equation Eq. (1).
 9. Identify and remove the edge with high betweenness score so that spanning subgraph SG gets segmented into 2 clusters.
 10. Repeat steps 9 and 10 until we get k clusters.
-

subgraph G is that it takes few iterations to get k modules. Description of the proposed approach is summarized in Algorithm 1. Complexity of the proposed algorithm is $O(n^3)$.

3 Experimental Analysis

We compare performance of the proposed algorithm against two recent graph based clustering algorithms namely MST-based clustering algorithm using Betweenness heuristics (B-MST) [12], Eigenanalysis on MST-based neighborhood graph (E-MST). The algorithms are evaluated using both artificial as well as real cancer cell datasets. The clusters generated by different algorithms are evaluated using Adjusted Rand Index (ARI) and Silhouette Index (SI). ARI is an external cluster validity index that measures the degree of agreement between two partitions, generally ground truth partitions (P_1) and predicted partitions (P_2). ARI score ranges from -1 to 1 , where the value -1 indicating complete disagreement and the value 1 indicating complete agreement between P_1 and P_2 . Silhouette index represents the degree of cohesion and separation of clusters obtained by a clustering algorithm. The silhouette index takes values in the range 0 to 1 , where the value 1 indicates all the objects are appropriately clustered.

First, performance of the proposed algorithm is illustrated using two artificial datasets, namely flame and path-based dataset. Flame dataset contains two clusters: a spherical shaped and a half-ring shaped cluster. Path-based dataset consists of two Gaussian clusters and one open ring cluster surrounding the

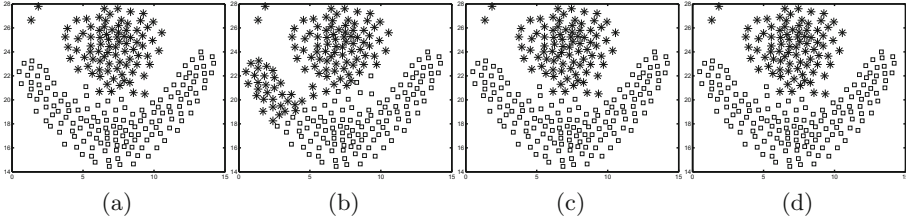


Fig. 2. Comparison of algorithms on flame dataset. (a) Dataset. (b, c, d) Clusters identified by B-MST [12], E-MST [9] and proposed algorithm.

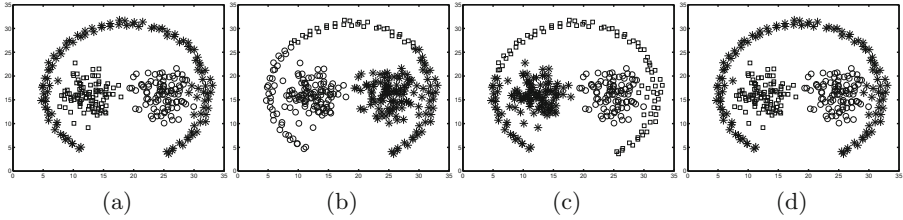


Fig. 3. Comparison of algorithms on path-based dataset. (a) Dataset. (b, c, d) Clusters identified by B-MST [12], E-MST [9] and proposed algorithm.

first two. Figures 2 and 3 show performance of the proposed algorithm on these datasets. The figures indicate that the proposed algorithm retrieves the expected clusters.

For real datasets, we consider 5 cancer cell datasets, namely BreastA, BreastB, DLBCLA, Novartis and ALB taken from Broad Cancer Institute repository (<http://broadinstitute.org/cgi-bin/cancer>). BreastA dataset comprises of 98 samples which are grouped into three classes having 11, 51 and 36 samples. Similarly, BreastB dataset includes 49 samples from two different classes of estrogen receptor. The diffuse large B-cell lymphoma A, or DLBCLA gene dataset has 141 samples with 661 attributes. The dataset is partitioned into three classes. Novartis dataset comprises of 103 objects from four different types of cancer namely 26 breast, 26 prostate, 28 lung and 23 colon tissues. Each object in this dataset has 1000 attributes. ALB dataset corresponds to bone marrow samples obtained from acute leukemia patients at the time of diagnosis. It contains 38 samples each having 1000 attributes. Table 1 shows that the proposed algorithm obtains higher values of ARI and SI as compared to other algorithms almost on all the datasets. This indicates that the proposed algorithm achieves both inter-cluster separation as well as within-cluster cohesion. Cancer gene expression datasets are highly heterogeneous comprising clusters of different shapes and sizes and with varying levels of overlapping. But the proposed algorithm is able to retrieve such clusters by effectively identifying the bottle neck edges from the spanning subgraph. This enabled the algorithm to achieve highest scores of ARI and SI as compared to other algorithms B-MST and E-MST.

Table 1. Comparison of ARI and Silhouette Index (SI) obtained by different algorithms on Cancer cell datasets. Highest score is marked in bold.

Dataset	ARI			SI		
	B-MST [12]	E-MST [9]	Proposed	B-MST [12]	E-MST [9]	Proposed
Breast-A	0.5654	0.7122	0.8807	0.2487	0.3689	0.7533
Breast-B	0.1556	0.4322	0.7134	0.1783	0.2587	0.4589
DLBCLA	0.1628	0.2313	0.7155	0.2444	0.2332	0.3581
Novartis	0.9460	0.8308	0.9299	0.4145	0.4587	0.7480
ALB	0.7810	0.9113	0.9203	0.3987	0.4394	0.8256

Table 2. Comparison of modularity metric (Q) obtained by different algorithms on Cancer cell datasets. Highest score is marked in bold.

Dataset	B-MST [12]	E-MST [9]	Proposed
Breast-A	0.2879	0.3011	0.4089
Breast-B	0.1765	0.2573	0.3822
DLBCLA	0.3086	0.3800	0.4587
Novartis	0.6145	0.6357	0.6266
ALB	0.3416	0.372	0.4265

Table 3. Number of iterations of edge removal (I) taken by the proposed algorithm. Lowest score is marked in bold.

Dataset	B-MST [12]	Proposed
Breast-A	23	15
Breast-B	16	12
DLBCLA	19	17
Novartis	11	7
ALB	8	5

We also assess the clusters obtained by different clustering algorithms using Newman’s modularity metric, which is a measure of quality of partitioning of a network into modules (clusters or communities) [11]. The modularity is computed as $Q = \sum_i (e_{ii} - a_i^2)$, where e_{ii} is intra-module edges of i^{th} -module; a_i is the fraction of edges in the network that connects nodes in module i to other modules in the network. Table 2 presents the modularity index obtained by different algorithms. The modularity score of the proposed algorithm indicates that it attains a relatively quality partitioning than other two algorithms. We have also recorded the number of iterations of betweenness computation taken by the proposed algorithm and the values are shown in Table 3. The proposed algorithm takes fewer iterations as compared to B-MST [12] algorithm.

4 Conclusion and Future Scope

This paper presents an empirical study of graph based clustering algorithms for identifying cancerous cell identification from microarray gene expression datasets. We have also proposed an improved clustering algorithm using betweenness on spanning subgraph. The proposed algorithm effectively identifies the cluster separation through betweenness centrality. Experimental results indicate that the proposed algorithm has shown improvement on cancer cell identification as compared to other two clustering algorithms.

References

1. Bayá, A.E., Granitto, P.M.: Clustering gene expression data with a penalized graph-based metric. *BMC Bioinform.* **12**(1), 2–19 (2011)
2. Bayá, A.E., Larese, M.G., Granitto, P.M.: Clustering using PK-D: a connectivity and density dissimilarity. *Expert Syst. Appl.* **51**(1), 151–160 (2016)
3. Dost, B., Wu, C., Su, A., Bafna, V.: TCLUS: a fast method for clustering genome-scale expression data. *IEEE/ACM Trans. Comput. Biol. Bioinform. (TCBB)* **8**(3), 808–818 (2011)
4. Hoshida, Y., Brunet, J.P., Tamayo, P., Golub, T.R., Mesirov, J.P.: Subclass mapping: identifying common subtypes in independent disease data sets. *PLoS ONE* **2**(11), e1195 (2007)
5. Huttenhower, C., Flamholz, A.I., Landis, J.N., Sahi, S., Myers, C.L., Olszewski, K.L., Hibbs, M.A., Siemers, N.O., Troyanskaya, O.G., Collier, H.A.: Nearest Neighbor Networks: clustering expression data based on gene neighborhoods. *BMC Bioinform.* **8**(250), 1–13 (2007)
6. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. *ACM Comput. Surv. (CSUR)* **31**(3), 264–323 (1999)
7. Jay, J.J., Eblen, J.D., Zhang, Y., Benson, M., Perkins, A.D., Saxton, A.M., Voy, B.H., Chesler, E.J., Langston, M.A.: A systematic comparison of genome-scale clustering algorithms. *BMC Bioinform.* **13**(Suppl 10), S7 (2012)
8. Jiang, D., Tang, C., Zhang, A.: Cluster analysis for gene expression data: a survey. *IEEE Trans. Knowl. Data Eng.* **16**(11), 1370–1386 (2004)
9. Jothi, R., Mohanty, S.K., Ojha, A.: Functional grouping of similar genes using eigenanalysis on minimum spanning tree based neighborhood graph. *Comput. Biol. Med.* **71**, 135–148 (2016)
10. Jothi, R., Mohanty, S.K., Ojha, A.: Fast approximate minimum spanning tree based clustering algorithm. *Neurocomputing* **272**, 542–557 (2017)
11. Newman, M.E.: Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* **74**(3), 036104 (2006)
12. Pirim, H., Ekşioğlu, B., Perkins, A.D.: Clustering high throughput biological data with B-MST, a minimum spanning tree based heuristic. *Comput. Biol. Med.* **62**, 94–102 (2015)
13. Ruan, J., Dean, A.K., Zhang, W.: A general co-expression network-based approach to gene expression analysis: comparison and applications. *BMC Syst. Biol.* **4**(1), 8 (2010)
14. de Souto, M.C., Costa, I.G., de Araujo, D.S., Ludermir, T.B., Schliep, A.: Clustering cancer gene expression data: a comparative study. *BMC Bioinform.* **9**(1), 1–14 (2008)
15. Thalamuthu, A., Mukhopadhyay, I., Zheng, X., Tseng, G.C.: Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics* **22**(19), 2405–2412 (2006)
16. Xu, R., Wunsch, D.C.: Clustering algorithms in biomedical research: a review. *IEEE Rev. Biomed. Eng.* **3**, 120–154 (2010)
17. Yu, Z., Wong, H.S., Wang, H.: Graph-based consensus clustering for class discovery from gene expression data. *Bioinformatics* **23**(21), 2888–2896 (2007)

MahalCUSFilter: A Hybrid Undersampling Method to Improve the Minority Classification Rate of Imbalanced Datasets

Venkata Krishnaveni Chennuru^(✉) and Sobha Rani Timmappareddy^(✉)

SCIS, University of Hyderabad, Hyderabad, India
cvkrishnaveni19@gmail.com, tsracs@uohyd.ernet.in

Abstract. Class Imbalance problem has received considerable attention in the machine learning research. Among the methods which handle class imbalance problem, *undersampling* is a data level approach which preprocesses the data set to reduce the size of the majority class instances. Most of the existing undersampling methods apply either prototype selection or clustering techniques to balance the data set. They are effective and popular, but both processes are complex. Drawbacks of the cluster based undersampling methods are: The quality of the chosen majority class samples varies depending on clustering algorithm, number of clusters and also the convergence is difficult. Drawback of prototype selection methods is that they have to compare each majority instance with its k nearest neighbors to decide which majority class instance should be selected/discarded which is not only time consuming and is also difficult to implement for large datasets. Proposed undersampling method **MahalanobisCentroidbasedUndersamplingwithFilter (MahalCUSFilter)** overcomes the above said problems: *parameter dependence, complexity and information loss*. Proposed method is used in conjunction with c4.5 and kNN classifiers, and found to improve the minority class classification rate of all datasets with comparable overall performance for the entire dataset. To the best of our knowledge this kind of grouping has not been used in undersampling to improve the classification accuracy of imbalanced data sets.

1 Introduction

In many real world applications cardinality of rarer cases (minority/positive) is much smaller than cases that are common(majority/negative). For example, data pertaining to people affected by diseases like cancer (minority) are rare and rest of the people are normal (majority). A data set having this type of property is called an *Imbalanced* data set. Traditional classifiers do not perform well on imbalanced data sets. Main reasons for this are the inherent assumptions made for traditional classifiers such as equal class distribution in the training set, equal mis-classification cost for both the classes and the performance dependency on accuracy. Overall classification rate of test set is computed irrespective of their class distribution which results in giving higher accuracy even if all minority

instances are misclassified. Minority cases of that rare-disease are only 2% and the rest are 98%. Classifying all unseen cases as majority class would give an accuracy of 98%, even while misclassifying all minority cases as majority, which is a major problem with classification of imbalanced datasets.

Several solutions are proposed at data level and algorithmic level to handle the problem of imbalance in the data sets and to reduce the impact of imbalance on the minority instances classification. Cost-sensitive learning, ensemble methods are also widely implemented to address this problem. At data level, the data set is manipulated to balance the class distribution. Data level sampling methods for handling imbalanced data sets are categorized into (i) Oversampling methods and (ii) Undersampling methods. At algorithmic level, thresholds and parameters in the algorithm are adjusted in classification methods to handle the imbalance. In the case of Cost-Sensitive learning, cost matrix with unequal costs, more penalty for false negatives and low penalty for false positive is used. Ensemble learning methods make use of subsets of the samples of the data set and several classifiers to improve the classification of imbalanced data sets. Proposed method in this work comes under undersampling category.

Section 2 covers related literature on imbalanced data sets. Section 3 provides framework of the proposed methods, Sect. 3.3 describes experiments and results, Sect. 3.4 provides discussion through comparison. Section 4 gives the conclusions.

2 Related Work

In the literature, several papers provide a very good survey on the classification of imbalanced data sets and on various methods which can handle the imbalance problem.

2.1 Undersampling

Undersampling methods reduce the number of majority class instances to balance the data set. These can be divided into Random Undersampling and Informative Undersampling. Random undersampling, removes majority instances randomly to balance the training set. Because of this, there is loss of useful information. Informative undersampling, chooses or discards certain majority instances based on certain conditions. Many solutions are proposed based on informative undersampling. It can be surmised that most of the methods in undersampling deal with either kNN based approaches, clustering based approaches or a combination of these two approaches.

Popular Undersampling Techniques. Condensed Nearest Neighbor (CNN) Rule [8], the Condensed Nearest Neighbor Rule with Tomek Link (CNNTL) [5], Neighborhood Cleaning Rule (NCL) [10], One Sided Selection (OSS) [9], Tomek Link [18] etc. are widely used undersampling techniques. They select majority class samples based on their distance from minority class samples using kNN classifier.

Rushi et al. [11] clustered majority class samples X_{Maj} into k clusters and select $R_i \times \text{size}(\text{MinClass})$ number of samples from each cluster so that the total number of selected majority samples equals the size of the minority set X_{Min} to balance the training set, where $R_i = X_{Maj_i}/X_{Maj}$, $1 \leq i \leq k$ represents the number of majority class samples to be chosen is based on the ratio of the number of majority samples in each cluster to the total number of majority samples. Number of majority class samples chosen from i th cluster is calculated using $S_i = X_{Min} \times R_i$, $1 \leq i \leq k$. In [16], a cluster based undersampling with ensemble learning is proposed. The authors have clustered the majority instances into k clusters where k value lies between 1 and size of the minority class and $\text{size}(\text{MinClass})/k$ number of samples are selected from each cluster to be equal to the number of minority samples. m training sets are formed and trained using m classifiers and the final result is obtained by weighted majority voting, where weight of each classifier is taken as the inverse of its error on the whole training set. In [14], majority class instances are divided into k clusters and k training sets are formed with majority class instances of each cluster combined with all the minority class samples and the training sets with highest accuracy is used as the final training set for classification.

Latest papers on imbalanced data sets include [4, 6, 12, 13, 19, 22] etc. Wang et al. in [19] use an ensemble method along with weights and information about sample misclassification to effectively classify imbalanced data. Zhang et al. [22] present empirical analysis by conducting various experiments on imbalanced data sets of varying imbalance, size and complexity applying three popular classifiers Naive Bayes, c4.5 and SVM. Results have shown that SVM outperforms the other two classifiers. Barella et al. in [4] proposed a cluster based one sided selection method for undersampling. In [6], a similarity based hierarchical decomposition method is proposed to classify imbalanced data sets. Wing et al. in [13] proposed a diversified sensitivity-based undersampling method for imbalance classification. Another latest works, [12] uses ensembles of First Order logical Decision Trees to handle the problem of imbalanced classification, [2] uses feature weighting to deal with overlap in imbalanced datasets, [7] proposed a RandomBalance method for imbalanced data which uses ensembles of variable priors classifiers. In [17], Sun et al. proposed a novel ensemble method to classify imbalanced data sets.

3 Proposed Method

3.1 Motivation for the Proposed Method

Existing under sampling methods to balance the imbalanced data set either apply

- **Prototype Selection Methods**

Drawback: Complexity involved in choosing the majority class samples is high since selection is done based on the distance from k nearest neighbors that

is, every majority sample is to be compared with k nearest neighbors which is arduous. Complexity and time consumption of the method increases with the increase in number of instances or number of attributes of the dataset.

– Clustering Algorithms

Once clusters are formed, this method is simple to implement but to form clusters several issues are to be addressed viz. (i) Which clustering algorithm is to be used? This decision depends mostly on the size, dimension and type of the dataset. (ii) How many clusters are to be formed? This can be decided by using cluster validity indices. Again in those, if external cluster validity indices are chosen, parameters are to be supplied by the user i.e., again quality of the cluster may vary depending upon the parameters. Even, if internal cluster validity indices are used, which is appropriate and why are to be known.

3.2 Theoretical Background of Proposed Method

All the real world datasets, are multivariate in nature. So, a distance measure for multivariate dataset should consider not only the variances of the attributes but also their covariances or correlations. The Euclidean distance measure between two vectors is not helpful in some situations as adjustment for the variances or the covariances is not possible. So a statistical distance, a standardized measure first proposed by Mahalanobis in 1936, often referred to as Mahalanobis distance is considered. A random variable with larger variance receives relatively less weight than others in a Mahalanobis distance. Two highly correlated variables do not contribute more than two less correlated variables in case of mahalanobis distance. The essence of Mahalanobis distance measure is to use the inverse of the covariance matrix which has the effect of standardizing all variables to the same variance and eliminates the correlations [15]. Mahalanobis distance is found using d_{Mahal} (Eq. 1)

$$d_{Mahal} = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})} \quad (1)$$

The idea behind MahalanobisCentroid based UnderSampling with Filter (MahalCUSFilter) method is to capture the majority class samples based on their similarity with respect to the average behaviour of all the majority class instances. Mahalanobis distance metric is considered here to compute the distances between the reference point (Mean-Vector i.e., Centroid) and each majority class instance in the data set. This method chooses the majority class samples based on their distance from their Mean-Vector (centroid(C_{cent})), that is samples throughout the distribution from nearer to farther distances so as to reflect the distribution of the majority class instances As a last step, majority class instances close to minority class instances are filtered out and the remaining training set is used for training. This will try to ensure to have a clear boundary between minority and majority class instances by eliminating risky instances. Steps followed in the pre-processing are shown in the Algorithm 1. Number of groups are chosen arbitrarily.

This method addresses three major drawbacks of existing undersampling methods:

Parameter Dependence: MahalCUSFilter performance does not depend on the parameters set by the user, unlike cluster based and kNN based undersampling methods whose performance depend on the chosen clustering algorithm, chosen number of clusters, k-value etc. Experimentation with different number of bins do not show much variation in the results obtained.

Complexity: MahalCUSFilter is very simple to implement, unlike other methods which take time to converge, and whose complexity increases with increased number of dimensions and instances in the dataset.

Information loss: The issue of representation of the majority class is addressed by employing stratified sampling strategy, which chooses the number of samples from each group depending on its size, so that the samples chosen represent the majority class as a whole.

Novel features of this method compared to other existing popular and latest methods are discussed here. The main difference between [11, 16, 21] and the proposed method is that they form groups using clustering whereas the proposed methods do not use any clustering algorithm, instead they divide the majority class samples based on their distance from the reference point.

3.3 Framework for the Proposed Method

Five-fold cross-validation is used in the experimentation. The new balanced training sets obtained after pre-processing using MahalCUSFilter are used to train the classifiers. Average of the results obtained on 5 test sets is shown as the output of the classifier. Two classifiers, C4.5 and kNN are chosen to check the performance of the proposed method.

MahalCUSFilter divides all the majority class instances into m bins based on the distance from their Centroid (Neg_{cent}). As a second step, (N_p) number of negatives are chosen using stratified sampling, that means number of instances to be chosen from each bin depends the size of the group and the total number of majority class instances to be chosen from all the groups which is equal to the number of minority class instances in the training set. This method is implemented and tested using classifiers C4.5 and kNN and compared with other existing undersampling techniques using the same classifiers.

Let N_p and N_n be the number of *Minority* class and *Majority* class instances of the training set respectively. After the formation of the groups, N_p *Majority* instances are selected based on stratified sampling to balance the minority and majority instances. r_i instances are chosen in each group so that resultant number of chosen *Majority* class samples will be equal to *Minority* class samples.

$$r_i = \frac{size(group_i)}{N_n} N_p \quad 1 \leq i \leq k \quad \text{Negative samples chosen} = \Sigma_i r_i \quad (2)$$

Algorithm 1. Mahalanobis Centroid based UnderSampling with Filter*Input:* An Imbalanced Data Set*Output:* A Balanced Data Set

1. Find the Mean-Vector(Centroid) of the majority class instances of the training set, Neg_{cent} .
2. Find the *Mahalanobis* distance of each majority class training instance, i from Neg_{cent} . Let it be $dist(i)$.
3. Normalize the values of $dist(i)$ to the range 0 to 1.
4. The majority class training instances which are at a distance from 0 to 0.1 are placed in group1, 0.1 to 0.2 in group2 and so on 0.9 to 1.0 in group10.
5. r (Eq. 2) samples are chosen from each group to select a total of N_p (Number of Minority Class instances) Majority class instances.
6. The set with all minority class and chosen majority class from the given training set form a balanced set *New – Training*.
7. The misclassified majority class instances with 1NN(Filter) are filtered out from the new training set.

Evaluation Criteria. Performance of a classifier is calculated based on the confusion matrix. Various measures used for describing the performance of the classifiers are based on the confusion matrix and are listed below. Here TP: True positives, TN: True negatives, FP: False positives and FN: False negatives.

$$Sensitivity = TP_Rate = Recall = \frac{TP}{TP+FN}$$

$$Specificity = TN_Rate = \frac{TN}{TN+FP}$$

$$FP_Rate = \frac{FP}{TP+FN}$$

$$FN_Rate = \frac{FN}{TN+FP}$$

$$Accuracy = \frac{(TP+TN)}{(TP+FN+TN+FP)}$$

$$Error_Rate = \frac{(FP+FN)}{(TP+FN+TN+FP)}$$

Table 1. Details of the data sets.

Name of the data set	# Features	Total # instances	Imbalance ratio
Ecoli4	7	459	14.3
Haberman	3	306	2.78
Iris0	4	150	2
LibrasMove	90	360	14
NewThyroid1	5	215	5.14
Pima	8	768	1.89
Scene	294	2407	12.6
Spectrometer	93	7797	10.8
Yeast1289Vs7	8	947	30.57

Table 2. Comparison of Sensitivity, GMean and Balanced Accuracy results with **c4.5** classifier with Unprocessed Original training set, MahalCUSFilter and other popular undersampling methods.

Data set	Measure	Original	MahalCUS Filter	CNN (1968)	CNNTL (2004)	CPM (2005)	NCL (2001)	OSS (1997)	TL (1976)
Ecoli4	Sensitivity	0.56	0.94	0.75	0.85	0.70	0.65	0.80	0.65
	GMean	0.75	0.87	0.83	0.85	0.81	0.80	0.84	0.80
	Balanced-Accuracy	0.77	0.87	0.83	0.85	0.81	0.81	0.84	0.81
Haberman	Sensitivity	0.40	0.48	0.54	0.72	0.46	0.74	0.56	0.46
	GMean	0.57	0.57	0.62	0.58	0.59	0.62	0.63	0.61
	Balanced-Accuracy	0.62	0.60	0.63	0.60	0.61	0.63	0.64	0.63
Iris0	Sensitivity	0.98	0.98	0.94	0.94	0.80	0.98	0.94	0.98
	GMean	0.99	0.99	0.97	0.97	0.69	0.99	0.97	0.99
	Balanced-Accuracy	0.99	0.99	0.97	0.97	0.70	0.99	0.97	0.99
NewThyroid1	Sensitivity	0.91	0.91	0.94	0.97	0.94	0.91	0.94	0.86
	GMean	0.95	0.93	0.94	0.92	0.81	0.94	0.94	0.92
	Balanced-Accuracy	0.95	0.93	0.94	0.92	0.82	0.94	0.94	0.92
Pima	Sensitivity	0.62	0.77	0.76	0.89	0.51	0.85	0.78	0.69
	GMean	0.70	0.74	0.72	0.62	0.65	0.70	0.66	0.71
	Balanced-Accuracy	0.71	0.74	0.72	0.66	0.67	0.72	0.67	0.71
Spectrometer	Sensitivity	0.74	0.83	0.84	0.89	0.82	0.76	0.84	0.78
	GMean	0.85	0.84	0.81	0.79	0.83	0.85	0.81	0.87
	Balanced-Accuracy	0.86	0.84	0.81	0.80	0.83	0.86	0.81	0.88
Yeast1289vs7	Sensitivity	0.24	0.57	0.20	0.27	0.27	0.07	0.23	0.10
	GMean	0.42	0.51	0.45	0.51	0.52	0.26	0.48	0.31
	Balanced-Accuracy	0.62	0.59	0.60	0.62	0.63	0.53	0.61	0.54
LibrasMove	Sensitivity	0.63	0.88	0.88	0.79	0.58	0.58	0.79	0.67
	GMean	0.78	0.79	0.84	0.78	0.70	0.75	0.79	0.81
	Balanced-Accuracy	0.80	0.80	0.84	0.78	0.71	0.78	0.79	0.83
Scene	Sensitivity	0.23	0.61	0.47	0.55	0.35	0.28	0.41	0.24
	GMean	0.47	0.61	0.59	0.59	0.54	0.51	0.30	0.47
	Balanced-Accuracy	0.59	0.61	0.61	0.59	0.50	0.60	0.32	0.59

$$Precision = \frac{TP}{TP+FP}$$

$$Gmean = \sqrt{Sensitivity \times Specificity}$$

$$Balanced Accuracy = \frac{Sensitivity+Specificity}{2}$$

Details of the Datasets Used in Experiments. Binary class data sets are used in the experiments. Multi-class data sets are converted into binary class

Table 3. Comparison of Sensitivity, GMean and Balanced Accuracy results with kNN($k = 1$) classifier with Unprocessed Original training set, MahalCUSFilter and other popular undersampling methods.

Data set	Measure	Original	MahalCUS Filter	CNN (1968)	CNNTL (2004)	CPM (2005)	NCL (2001)	OSS (1997)	TL (1976)
Ecoli4	Sensitivity	0.69	1.00	0.85	0.85	0.40	0.75	0.80	0.75
	GMean	0.82	0.89	0.91	0.90	0.63	0.85	0.88	0.86
	Balanced- Accuracy	0.84	0.89	0.91	0.90	0.69	0.86	0.88	0.87
Haberman	Sensitivity	0.50	0.53	0.46	0.69	0.36	0.68	0.53	0.51
	GMean	0.53	0.54	0.54	0.54	0.50	0.56	0.53	0.58
	Balanced- Accuracy	0.53	0.55	0.54	0.56	0.53	0.57	0.53	0.58
Iris0	Sensitivity	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	GMean	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Balanced- Accuracy	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
NewThyroid1	Sensitivity	0.97	1.00	0.97	1.00	0.91	1.00	0.97	0.97
	GMean	0.98	0.99	0.97	0.98	0.95	0.99	0.97	0.98
	Balanced- Accuracy	0.98	0.99	0.97	0.98	0.95	0.99	0.97	0.98
Pima	Sensitivity	0.53	0.69	0.64	0.89	0.54	0.83	0.84	0.74
	GMean	0.65	0.70	0.65	0.56	0.64	0.70	0.68	0.73
	Balanced- Accuracy	0.67	0.70	0.65	0.62	0.65	0.71	0.70	0.73
Spectrometer	Sensitivity	0.74	0.85	0.87	0.89	0.84	0.82	0.80	0.78
	GMean	0.85	0.90	0.91	0.90	0.91	0.90	0.88	0.88
	Balanced- Accuracy	0.86	0.91	0.92	0.90	0.91	0.90	0.88	0.88
Yeast1289vs7	Sensitivity	0.14	0.70	0.33	0.67	0.27	0.23	0.47	0.13
	GMean	0.32	0.66	0.51	0.65	0.50	0.47	0.61	0.36
	Balanced- Accuracy	0.95	0.65	0.56	0.65	0.60	0.59	0.64	0.55
LibrasMove	Sensitivity	0.71	0.67	0.75	0.92	0.75	0.88	0.75	0.71
	GMean	0.84	0.80	0.85	0.93	0.86	0.92	0.85	0.84
	Balanced- Accuracy	0.85	0.81	0.86	0.93	0.87	0.92	0.86	0.85
Scene	Sensitivity	0.17	0.60	0.37	0.63	0.25	0.34	0.44	0.24
	GMean	0.41	0.59	0.54	0.59	0.47	0.55	0.58	0.47
	Balanced- Accuracy	0.56	0.59	0.58	0.59	0.56	0.61	0.60	0.58

by taking required class as minority class and all other classes as majority class instances. The data sets are chosen based on their number of attributes, number of instances and imbalance ratio to check the performance of the proposed methods on small, medium and large number of attributes, instances and imbalance ratio. Table 1 shows the details of the data sets. All the values in these data sets are numeric except the class attribute. The data sets are taken from the UCI [3],

KEEL [1] data repositories and used KEEL [1], WEKA [20] tools to conduct the experiments.

3.4 Results

Proposed MahalCUSFilter method is compared with other undersampling methods like **CNN**, **CNNTL**, **CPM**, **OSS**, **TL** which are popular among the undersampling methods in the literature as it belongs to undersampling method. Difference between the proposed method and the other methods is in the way the majority class instances are chosen. They choose/discard the majority class samples based on their distance from minority class samples whereas MahalCUSFilter choose majority class samples based on their distance from the reference point (Centroid(*Neg_Cent*)). They use kNN classifier to choose majority class samples, but the proposed methods do not use any classifier thereby reducing computation involved in finding nearest neighbors as the size of the data set increases. Results from Table 2, prove that the proposed method outperform these methods in a few data sets and obtain comparable results with these methods in other remaining sets. In no case, the proposed methods are proven to be inferior to all these existing popular undersampling methods in classifying minority class instances (Table 3).

4 Conclusions

In this paper, MahalCUSFilter, a hybrid undersampling method is proposed to balance the training set by choosing the samples from the majority class equal to the number of minority class samples. Most of the existing undersampling methods apply either (a) prototype selection or (b) clustering techniques to balance the data set which are parameter dependent and convergence is difficult. The proposed methods are **simple** and **parameter independent**. These are based on distribution specific grouping, additionally, issues considered are: (i) Information loss and (ii) Proper representation of the majority class. Moreover, the methods are simple to implement and effective even for high dimensional and large data sets.

A good insight is obtained by these methods. Unlike in kNN or clustering methods, MahalCUSFilter partitions the data set from a global perspective. Distribution Specific Grouping of the data is a kind of annular spherical neighbourhood in the case of Euclidean distance based methods. Hence, the probability of selecting the samples from the annular neighbourhood is better compared to the clustering, where, a local neighbourhood defines the clusters and there is a probability of missing some disjuncts altogether. All the kNN based undersampling methods are looking at a very small neighbourhood ($K = 1, 3$ or 5). This may introduce lot of bias.

Empirical results also support the theory. Clearly, MahalCUSFilter improves minority class classification rate on all datasets compared to unprocessed original

imbalanced datasets. In case of few datasets, other popular undersampling methods comparison gave better results. But for those methods complexity of time and space are more compared to MahalCUSFilter method. The major advantage of our method is that it is simple to implement hence consumes less time as compared to other undersampling methods and that it is not dependent on any of the input parameters and works well with data sets having large instances, large features, small instances and small features as well. To the best of our knowledge the kind of grouping that was used in the proposed methods has not been used so far in undersampling to improve the classification of imbalanced data sets.

References

1. Alcalá-Fdez, J., Sanchez, L., Garcia, S., del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., et al.: Keel: a software tool to assess evolutionary algorithms for data mining problems. *Soft Comput.-A Fus. Found. Methodol. Appl.* **13**(3), 307–318 (2009)
2. Alshomrani, S., Bawakid, A., Shim, S.-O., Fernández, A., Herrera, F.: A proposal for evolutionary fuzzy systems using feature weighting: dealing with overlapping in imbalanced datasets. *Knowl.-Based Syst.* **73**, 1–17 (2015)
3. Asuncion, A., Newman, D.: Uci machine learning repository (2007)
4. Barella, V.H., Costa, E.P., Carvalho, A.C.P.L.F.: Clusterosr: a new undersampling method for imbalanced learning. In: *Brazilian Conference on Intelligent Systems, 3rd; Encontro Nacional de Inteligência Artificial e Computacional, 11th.* Universidade de São Paulo-USP (2014)
5. Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explor. Newsl.* **6**(1), 20–29 (2004)
6. Beyan, C., Fisher, R.: Classifying imbalanced data sets using similarity based hierarchical decomposition. *Pattern Recogn.* **48**(5), 1653–1672 (2015)
7. Díez-Pastor, J.F., Rodríguez, J.J., García-Osorio, C., Kuncheva, L.I.: Random balance: ensembles of variable priors classifiers for imbalanced data. *Knowl.-Based Syst.* **85**, 96–111 (2015)
8. Hart, P.: The condensed nearest neighbor rule (corresp.). *IEEE Trans. Inf. Theory* **14**(3), 515–516 (1968)
9. Kubat, M., Matwin, S., et al.: Addressing the curse of imbalanced training sets: one-sided selection. In: *ICML*, vol. 97, Nashville, USA, pp. 179–186 (1997)
10. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. In: Quaglini, S., Barahona, P., Andreassen, S. (eds.) *AIME 2001. LNCS (LNAI)*, vol. 2101, pp. 63–66. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-48229-6_9
11. Longadge, M.R., Dongre, M.S.S., Malik, L.: Multi-cluster based approach for skewed data in data mining. *J. Comput. Eng. (IOSR-JCE)* **12**(6), 66–73 (2013)
12. Manjula, M., Seeniselvi, T.: Ensembles of first order logical decision trees for imbalanced classification problems
13. Ng, W.W., Hu, J., Yeung, D.S., Yin, S., Roli, F.: Diversified sensitivity-based undersampling for imbalance classification problems. *IEEE Trans. Cybern.* **45**(11), 2402–2412 (2015)

14. Rahman, M.M., Davis, D.: Cluster based under-sampling for unbalanced cardiovascular data. In: Proceedings of the World Congress on Engineering, vol. 3, pp. 3–5 (2013)
15. Rencher, A.C.: *Methods of Multivariate Analysis*, vol. 492. Wiley, Hoboken (2003)
16. Sobhani, P., Viktor, H., Matwin, S.: Learning from imbalanced data using ensemble methods and cluster-based undersampling. In: Appice, A., Ceci, M., Loglisci, C., Manco, G., Masciari, E., Ras, Z.W. (eds.) NFMCP 2014. LNCS (LNAI), vol. 8983, pp. 69–83. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-17876-9_5
17. Sun, Z., Song, Q., Zhu, X., Sun, H., Xu, B., Zhou, Y.: A novel ensemble method for classifying imbalanced data. *Pattern Recogn.* **48**(5), 1623–1637 (2015)
18. Tomek, I.: Two modifications of CNN. *IEEE Trans. Syst. Man Cybern.* **6**, 769–772 (1976)
19. Wang, C., Hu, L., Guo, M., Liu, X., Zou, Q.: imDC: an ensemble learning method for imbalanced classification with mirna data. *Genet. Mol. Res.* **14**(1), 123–133 (2015)
20. Witten, I.H., Frank, E., Trigg, L.E., Hall, M.A., Holmes, G., Cunningham, S.J.: *Weka: practical machine learning tools and techniques with Java implementations* (1999)
21. Yen, S.-J., Lee, Y.-S.: Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Syst. Appl.* **36**(3), 5718–5727 (2009)
22. Zhang, S., Sadaoui, S., Mouhoub, M.: An empirical analysis of imbalanced data classification. *Comput. Inf. Sci.* **8**(1), 151 (2015)

Bezier Curve Based Continuous Medial Representation for Shape Analysis: A Theoretical Framework

Leonid Mestekiy¹ and B. H. Shekar²(✉)

¹ Lomonosov Moscow State University, Moscow, Russia
mestlm@mail.ru

² Mangalore University, Mangalore, Karnataka, India
bhshekar@gmail.com

Abstract. In this paper, we introduce a continuous medial representation (skeleton + radial function) to compute the description of a given bitmap image. The computational geometry based mathematical model is proposed to obtain the continuous medial representation unlike traditional algorithms which are used to estimate medial representation of bitmap image in a discrete/heuristic manner. The skeleton of the polygonal figure is represented by straight line control graph of a compound Bezier curve which results in simple and accurate description. The process of pruning is devised to eliminate the spurious branches which are quite often exist while processing shapes in real scenario and hence continuous skeleton regularization is achieved for its accurate representation.

Keywords: Skeleton · Medial axis · Radial function · Continuous representation · Bezier curve · Shape analysis

1 Introduction

Skeleton (or medial axis transformation) is a powerful and widely known shape descriptor used in shape based object representation and classification. The computations of skeletons are commonly found in many of the shape representation based computer vision problems [2]. Skeletons are used as a basis to feature generation and hence to determine the similarity between objects for classification purpose. Originally, the concept of skeleton is defined for continuous objects. The skeleton of a closed region in Euclidean plane is a locus of centers of maximum empty circles of the region. The circle is considered to be empty if all its internal points are internal points of the region. The concept of skeleton (the middle set of points) was introduced and investigated by Blum [1] and later found to be one of the most popular methods for shape representation and classification.

The radial function of the skeleton assigns to each point of the skeleton the radius of the empty circle centered at this point. The skeleton together with the radial function form the so-called medial representation of the closed region [8]. The problem relies in computation of medial representation of digital images.

In principle, we have seen two approaches to compute the skeleton of digital images. The first approach which is the most popular because of ease of implementation is called

as discrete [3, 4, 8]. It consists of a morphological transformation of the original image (Fig. 1a) and construction of new digital image (Fig. 1b), which can be regarded as a discrete skeleton. In this new bitmap image, skeleton is represented by discrete lines which are of one pixel wide. The discrete radial function is calculated through the distance transform [9] of a digital image (Fig. 1c). The discrete approach is implemented in different ways using notion of distance maps, thinning, etc.

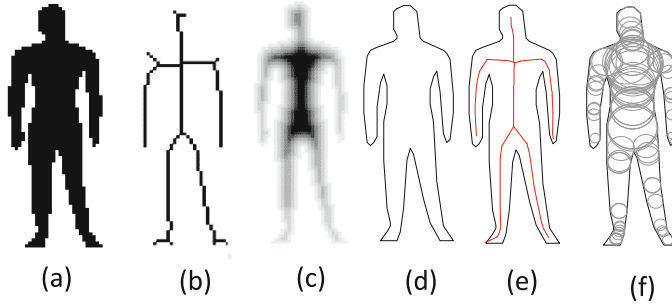


Fig. 1. (a) – The binary image, (b) – discrete skeleton, (c) – distance transform, (d) – polygonal figure, (e) – continuous skeleton, (f) – inscribed circles

Another approach to which this article is devoted, which we call continuous, is based on the approximation of a bitmap image by the geometrical figure in terms of a continuous geometry (Fig. 1d) and the construction of the skeleton (Fig. 1e) and radial function (Fig. 1f) for this figure. The resulting skeleton and radial function are considered as continuous medial representation of a digital bitmap image.

An important advantage of the continuous model is the ability to transform the image shape. The transformation of the image on the basis of a continuous medial representation is shown in Fig. 2. In this example, the digital object (Fig. 2a) is approximated (Fig. 2b) and a medial representation is constructed (Fig. 1e, f). The transformations consist of moving the points of the skeleton and changing the radial function at these points. New images (Fig. 2c–e) are constructed as the envelope of a family of modified circles. “Thick figure” (Fig. 2c) is obtained by multiplying the radial function (i.e., increasing the radii of the circles) by 1.2. “Athletic figure” (Fig. 2d) is obtained by reducing the radii of the circles corresponding to the lower half of the figure, by multiplying the radial function by 0.9. “Dancing figure” (Fig. 2e) is obtained by moving points and edges of the skeleton.

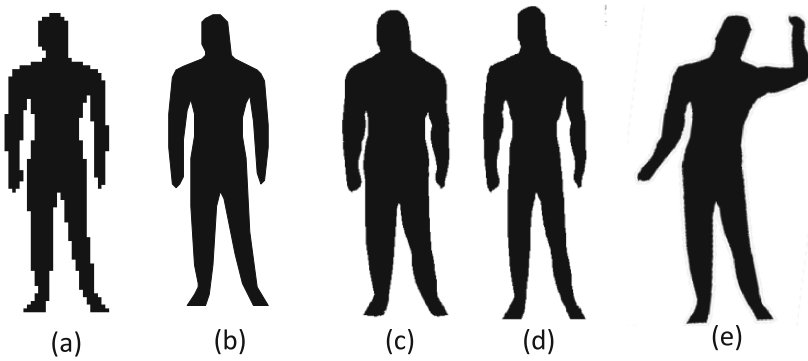


Fig. 2. Transformation of the object shape through continuous medial representation

In this article, we will show the general concept and our experience in developing and applying a continuous model for the medial representation of a digital image. The article generalizes the methods and algorithms developed earlier [7, 11].

2 Proposed Methodology

We proposed to obtain the medial representation (Fig. 1e, f) of a given a raster bitmap image (Fig. 1a) as follows:

- Approximation of a given binary raster image by a polygonal figure
- Construction of medial axes (skeleton) of the polygonal figure by computational geometry methods;
- Pruning of the skeleton – elimination of non-essential branches with the mathematically based criterion based on the Hausdorff metric;
- Description of the skeleton and radial function as a planar control graph of a compound Bézier curve.

The advantages of this approach are determined by the following factors.

1. Contours of objects are presented in the form of polygonal Figures
2. The skeleton is presented in the form of a geometric graph in which the vertices are points of the Euclidean plane, and the edges are straight lines or parabolic segments.
3. Each vertex of the skeleton graph is the center of the inscribed circle (e) and the radius and the tangency points on the figure boundary are known.
4. The inscribed circle centered at any point on the edges of the skeleton can be calculated
5. Coordinates and radial function for all points of the skeleton can be calculated with high accuracy in a floating format.
6. Continuous medial representation has the format of a flat geometric graph. For analysis, one can use effective algorithms of graph theory.

Let us consider in more detail the elements of the proposed approach.

3 Polygonal Approximation

We have considered polygons as the main model for approximating the boundaries of a digital image since the skeleton of a polygonal figure has a simple structure and can be obtained through efficient algorithms of computational geometry. The problem of approximation of a digital binary image is set as follows. Given a binary raster image (Fig. 3a), we need to build a polygonal shape (Fig. 3b).

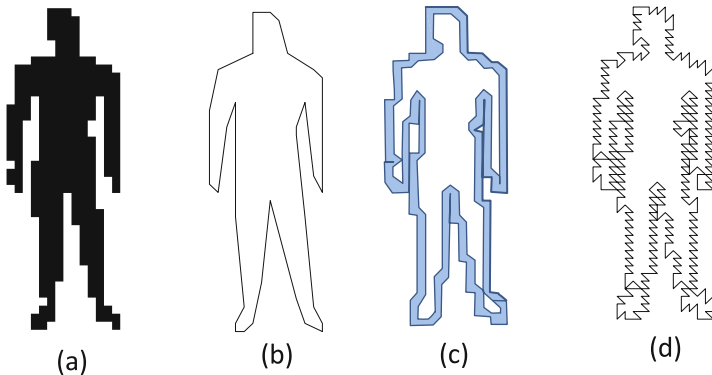


Fig. 3. (a) – The binary image, (b) – polygonal figure (minimal perimeter polygon), (c) – boundary corridors, (d) – tracing the border of a binary image

A polygonal figure must satisfy two requirements. Firstly, it must approximate the digital raster pre-image with high accuracy. Secondly, it must be topologically correct: boundary polygons should not have intersections and self-intersections. The proposed model ensures that these requirements are fulfilled. The model is based on the following principles:

- every pixel of the binary image is represented as black or white point in the center of pixel with integer coordinates on the Euclidean plane R^2 . The points of the object are black, the background points are white;
- pairs of adjacent multi-colored pixels form corridors that have one white and one black wall (Fig. 3c);
- each corridor is a ring in which the outer and inner boundary walls are polygons;
- any closed path inside the ring approximates the boundary of a discrete figure with an accuracy equal to the width of the corridor.

As a continuous model of the boundary of a discrete figure, we choose a closed rubber thread lying inside the corridor. The thread tends to assume a position at which its length is minimal. This position of the thread will be called the separating polygon of the minimal perimeter.

A polygonal figure formed with a rubber thread model satisfies the requirements stated above. It provides the best accuracy of approximation, because the rubber thread inside the corridor deviates from the black and white border pixels by no more than one

pixel. It is obvious that the dividing polygons of the minimal perimeter do not have self-intersections and do not intersect each other. The boundary corridor can be obtained by means of algorithms for tracing the boundary of a binary image. The result of such a trace is shown in Fig. 3d. The dividing polygon of the minimal perimeter within the corridor is a cyclic geodesic route inside the polygonal ring [10]. Effective algorithms for constructing geodesic paths in polygons are also well known.

4 Construction and Regularization of the Skeleton

The skeleton of a polygonal figure is defined as the set of centers of all circles inscribed in the figure (Fig. 4a, b). Effective algorithms have been developed for its construction based on the generalized Voronoi diagram [5, 6, 10].

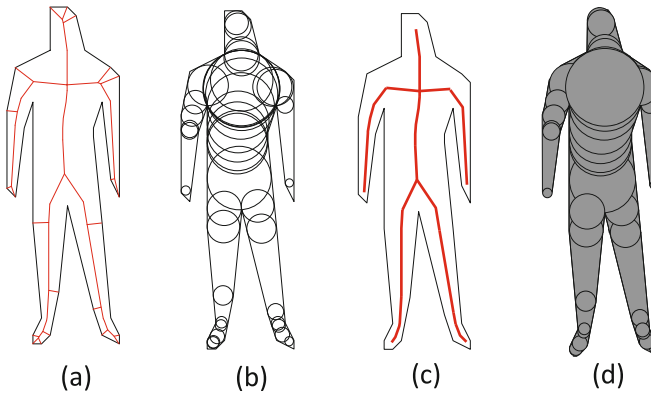


Fig. 4. (a) – The skeleton of polygonal figure, (b) – inscribed circles of polygonal figure, (c) – regularized skeleton, (d) – silhouette of a skeleton subgraph after pruning.

Polygons forming the boundary of a figure are divided into subsets of points, called sites. All the vertices of the figure form a set of sites-points, and all sides of the figure constitute a set of site-segments. Each site-segment is a side of the figure without its endpoints. For this set of sites, a generalized Voronoi diagram of linear segments is defined. This Voronoi diagram has the form of a flat graph. The skeleton is a subgraph of this graph (Fig. 4a).

The skeleton is a very sensitive object; this applies to both discrete and continuous skeletons. Sensitivity is expressed in the fact that the structure of the skeleton changes significantly as a result of variation of the boundary of the figure. In particular, the appearance of a small noise in the image leads to the appearance of numerous superfluous edges, which introduce confusion in the generation of features and generate errors in the recognition of the shape. In order to reduce the effect of noise, the constructed skeleton undergoes post-processing, which is called pruning. Pruning a skeleton removes a part of its branches which are classified as inessential and noisy (Fig. 4c). From the point of view of mathematics, this process is a regularization based on the correction of the solution obtained. The main problem of pruning is how to distinguish the “right”

branches of the skeleton from the “noise” branches? When pruning a discrete skeleton, this problem is solved on the basis of heuristic rules. The continuous skeleton of a polygonal shape provides an opportunity to solve this problem on the basis of a rigorous mathematically based model.

The medial representation allows us to represent the figure as the union of all inscribed circles with the centers on the skeleton (Fig. 4b). The pruning step is the removal of one edge of the skeleton and inscribed circles centered on this edge. The graph obtained as a result of removal of an edge is a subgraph of the source skeleton. The union of the rest inscribed circles with the centers on the subgraph is called the silhouette of the subgraph (Fig. 5). The silhouette of the subgraph is the subset of the polygonal figure. The criterion for removal of an edge is based on the Hausdorff distance between the silhouette and the figure. If this distance does not exceed a given threshold, then we remove an edge. The general rule of pruning is as follows. We remove all the terminal edges of the skeleton, for which the Hausdorff distance between the silhouette of the subgraph and the figure is less than a given threshold. An example of such pruning is shown in Fig. 4.

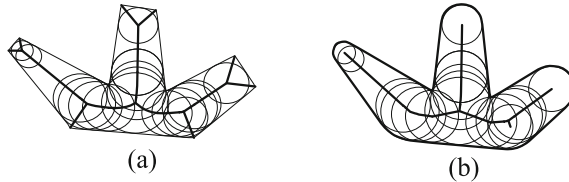


Fig. 5. (a) –Polygonal figure and its skeleton, (b) – truncated subgraph and its silhouette.

5 Bezier Curve Description

Each point of the continuous skeleton of a polygonal figure is the center of the inscribed circle. This makes it possible to set the radial function at the points of the skeleton as the radius of the circle centered at this point. As a result, we get a complete continuous medial representation for a discrete object: a skeleton and a radial function. However, the resulting “geometric” description (Fig. 6a) should be represented by an adequate data structure in a digital form. The main difficulty here is the description of the parabolic edges of the skeleton, as well as a description of the nonlinear radial function. A method for describing a skeleton and a radial function using rational Bézier splines of degree 1 and 2 is proposed in [11].

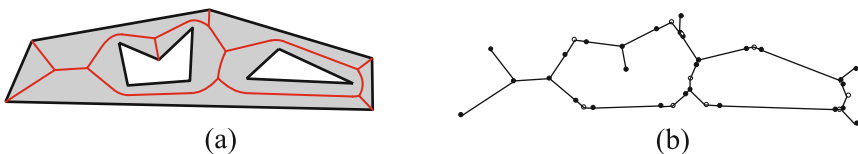


Fig. 6. (a) – Polygonal figure and its skeleton, (b) – control graph of the skeleton.

Thus, skeleton is a union of Bezier curves of first and second order. These curves describe the connected geometrical graph. We call this graph as the compound Bezier curve or Bezier curve graph. This Bezier curve graph can be represented by a straight-line geometric control graph (Fig. 6b). We call it as the control graph of the skeleton. Control graph has straight line edges. The set of control graph vertices includes all skeleton vertices and handle points of quadratic Bezier curves.

The control graph allows us in obtaining equations for all the edges of the skeleton and for the radial function. Thus, we obtain formulas for a complete description of the continuous medial representation. Each edge of the skeleton is a set of points equidistant from the two sites. Depending on the type of sites (v – point site, s – segment site) there are ss -edges, vv -edges and vs -edges in the skeleton. The ss -edge has the form of a straight line segment. Its shape and radial function are described as a Bezier curve of degree 1 with a parameter $t \in [0, 1]$:

$$x(t) = x_0 \cdot B_1^0(t) + x_1 \cdot B_1^1(t),$$

$$y(t) = y_0 \cdot B_1^0(t) + y_1 \cdot B_1^1(t),$$

$$r(t) = r_0 \cdot B_1^0(t) + r_1 \cdot B_1^1(t)$$

Here (x_0, y_0, r_0) and (x_1, y_1, r_1) are coordinates and radii of the end discs of the edge, $((x(t), y(t)))$ is the edge line, $r(t)$ is the radial function, $B_1^0(t) = (1 - t)$, $B_1^1(t) = t$ are Bernstein polynomials of degree 1.

The vs -edge is described through the Bézier splines of degree 2:

$$x(t) = x_0 \cdot B_2^0(t) + x_1 \cdot B_2^1(t) + x_2 \cdot B_2^2(t),$$

$$y(t) = y_0 \cdot B_2^0(t) + y_1 \cdot B_2^1(t) + y_2 \cdot B_2^2(t),$$

$$r(t) = r_0 \cdot B_2^0(t) + r_1 \cdot B_2^1(t) + r_2 \cdot B_2^2(t).$$

Here (x_0, y_0, r_0) and (x_2, y_2, r_2) are coordinates and radii of the end discs, (x_1, y_1, r_1) coordinates and radius of control disk, $B_2^0(t) = (1 - t)^2$, $B_2^1(t) = 2 \cdot (1 - t) \cdot t$, $B_2^2(t) = t^2$ – are Bernstein polynomials of degree 2.

A vv -edge is described by rational Bézier splines of degree 2:

$$x(t) = \frac{x_0 \cdot B_2^0(t) + w \cdot x_1 \cdot B_2^1(t) + x_2 \cdot B_2^2(t)}{B_2^0(t) + w \cdot B_2^1(t) + B_2^2(t)},$$

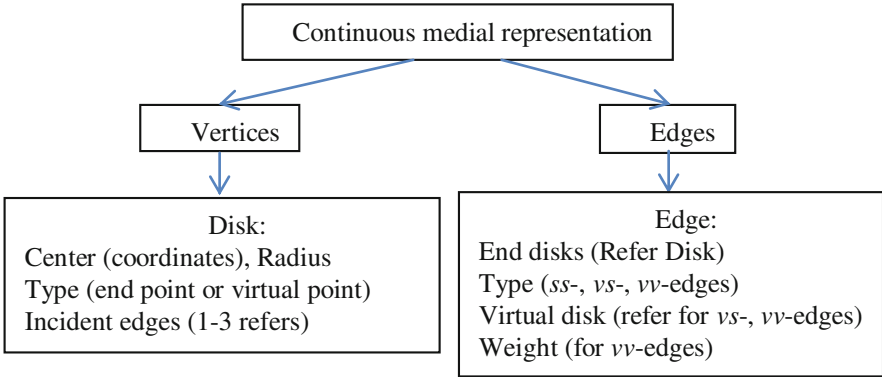
$$y(t) = \frac{y_0 \cdot B_2^0(t) + w \cdot y_1 \cdot B_2^1(t) + y_2 \cdot B_2^2(t)}{B_2^0(t) + w \cdot B_2^1(t) + B_2^2(t)},$$

$$r(t) = \frac{r_0 \cdot B_2^0(t) + w \cdot r_1 \cdot B_2^1(t) + r_2 \cdot B_2^2(t)}{B_2^0(t) + w \cdot B_2^1(t) + B_2^2(t)}.$$

Here all the parameters are the same as for the *vs*-edge, but an additional weight parameter *w* appears.

6 Data Structure

In the proposed approach, the heart of representation is a geometric graph. In the data structure, the graph is described by two arrays: an array of vertices and an array of edges. A disk is associated with each vertex of the graph which is described by the coordinates of the center and the radius. The edge of the graph is geometrically a segment of the line or curve, which is defined by a pair of end disks. With each edge, the parameters defining the radial function are connected. Depending on the shape of the edge and the type of radial function, there are three types of edges (*ss*-, *vs*-, *vv*-edges). The data structure describing control graph includes feature for each edge type. All edge descriptions contain the coordinates and radii of the two end disks - start (x_s, y_s, r_s) and finish (x_f, y_f, r_f). The *vs*-edge and *vv*-edge have kept additional parameters: coordinates and radius of control disk (x_c, y_c, r_c). The *vv*-edge has weight parameter *w*.



7 Experimental Results

In this section, we present the experimental results due to the proposed method and discuss one of the possible application. The proposed approach is applied to a problem of palm shape recognition. The medial representation is used to construct a measure of the difference in the images of the palms. This measure is used to classify the palms (Fig. 7).

The problem of comparing palms is reduced to the selection of such admissible transforms of them, in which their shapes will be the closest to one another. The difference of



Fig. 7. The silhouettes of palms (a) the first person, (b) the second person

their shapes in this (the closest) position is accepted as a measure of the distinction of objects.

Comparison of the palms is carried out after normalization. Normalization consists in bringing the palm to a standard position, in which the fingers are placed at predetermined angles to each other. The idea of normalization is shown in Fig. 8. On the basis of the medial representation (Fig. 8a) we find a “large” circle having the maximum radius among all inscribed circles of the palm. The points of intersection of a large circle with the branches of the skeleton are called joints (Fig. 8b). The skeleton determines the local coordinate system of the palm (Fig. 8c). The origin of coordinates is the center of a large circle, the direction of the abscissa axis coincides with the direction of the vector from the point of the bending of the middle finger (point 3) to the center of a large circle. As the unit of length along this axis, the radius of a large circle is selected. In this coordinate system, the palm is brought to the standard position by turning the skeleton axes (Fig. 8d) to the specified standard angles to the abscissa (130° , 160° , 180° , 200° , 230°). After performing this deformation, you need to calculate the silhouette of the normalized palm in the form of the envelope of the changed family of circles. Figure 9 shows examples of comparing the silhouettes of normalized images of the same palm and different palms.

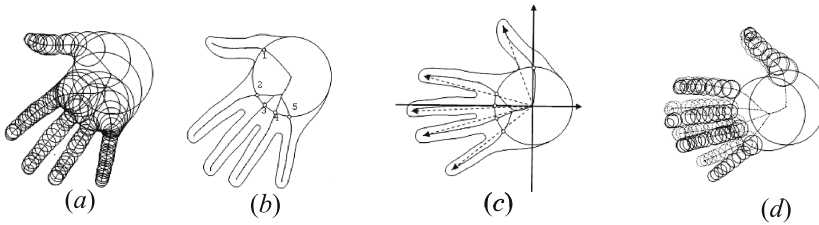


Fig. 8. Normalization of the image of the palm based on the transformation of the medial representation.



Fig. 9. Comparison of the silhouettes of the same palm and silhouettes of different palms.

8 Conclusion

The concept of continuous medial representation for a digital binary image is proposed in the article. It allows us to build a skeleton and radial function of a digital image in the form of a graph in which all the edges are represented by “lines with a width”. These lines are described using quadratic Bezier curves. The medial representation is described by the compound Bezier curve. The format of the description of compound Bezier curves using a rectilinear control graph makes it possible to construct a simple data structure that allows the creation of a medial representation with high accuracy for any digital image of arbitrarily complex shape.

Acknowledgements. This work is supported by Russian Foundation for Basic Research, RFBR Grant No. 16-57-45054 and Department of Science and Technology Grant No. INT/RUS/RFBR/P-248.

References

1. Blum, H., et al.: A transformation for extracting new descriptors of shape. In: *Models for the Perception of Speech and Visual Form*, vol. 19, no. 5, pp. 362–380 (1967)
2. da Fona Costa, L., Cesar Jr., R.M.: *Shape analysis and classification: theory and practice*. CRC Press, Inc. (2000)
3. Deng, W., Iyengar, S.S., Brenner, N.E.: A fast parallel thinning algorithm for the binary image skeletonization. *Int. J. High Perform. Comput. Appl.* **14**(1), 65–81 (2000)
4. Goutsias, J., Schonfeld, D.: Morphological representation of discrete and binary images. *IEEE Trans. Signal Process.* **39**(6), 1369–1379 (1991)
5. Lee, D.-T.: Medial axis transformation of a planar shape. *IEEE Trans. Pattern Anal. Mach. Intell.* **4**, 363–369 (1982)
6. Lee, D.-T., Drysdale III, R.L.: Generalization of Voronoi diagrams in the plane. *SIAM J. Comput.* **10**(1), 73–87 (1981)
7. Mestetskiy, L.M.: Skeletonization of a multiply-connected polygonal domain based on its boundary adjacent tree. *Sibirskii Zhurnal Vychislitel'noi Matematiki* **9**(3), 299–314 (2006)
8. Siddiqi, K., Pizer, S.M.: *Medial Representations: Mathematics, Algorithms and Applications*, vol. 37. Springer, Dordrecht (2008)
9. Strzodka, R., Telea, A.: Generalized distance transforms and skeletons in graphics hardware. In: *Proceedings of the Sixth Joint Eurographics-IEEE TCVG conference on Visualization*. Eurographics Association, pp. 221–230 (2004)
10. Chazelle, B.: A theorem on polygon cutting with applications. In: *Proceedings 23th IEEE Symposium on Foundations of Computer Science*, Chicago, pp. 339–349 (1982)
11. Mestetskiy, L.M.: Representation of segment Voronoi diagram by Bezier curves. *Program. Comput. Softw.* **41**(5), 279–288 (2015)

Trust Distrust Enhanced Recommendations Using an Effective Similarity Measure

Stuti Chug, Vibhor Kant^(✉), and Mukesh Jadon

Department of Computer Science and Engineering,
The LNM Institute of Information Technology, Jaipur, India
stutichug@gmail.com, vibhor.kant@gmail.com, jadonmukesh30@gmail.com

Abstract. Collaborative filtering (CF), the most prevalent technique in the area of recommender systems (RSs), provides suggestions to users based on the tastes of their similar users. However, the new user and sparsity problems, degrade its efficiency of recommendations. Trust can enhance the recommendation quality by mimicking social dictum “friend of a friend will be a friend”. However distrust, the another face of coin is yet to be explored along with trust in the area of RSs. Our work in this paper is an attempt toward introducing trust-distrust enhanced recommendations based on the novel similarity measure that combines user ratings and trust values for generating more quality recommendations. Our approach also exploits distrust links among users and analyses their propagation effects. Further, distrust values are also used for filtering more distrust-worthy neighbours from the neighbourhood set. Our experimental results show that our proposed approaches outperform the traditional CF and existing trust enhanced approaches in terms of various performance measures.

Keywords: Trust and distrust models · Recommender systems
Trust network · Collaborative filtering · Cold start and sparsity problem

1 Introduction

Due to the unprecedented proliferation of information available on the web, it is very difficult for users to find the relevant information from a large collection of data available online. To overcome the problem of information overload, web personalization tool would be the most prevalent tool. Recommender system (RS), a web personalization tool provides relevant suggestions to users based on their preferences [7]. The suggestions provided are aimed to support the decision-making process of users in various fields like videos, music, movies (MovieLens, Netflix), restaurants (Entree), books (Amazon), jokes (Jester). Many filtering techniques are used to construct RS such as content based filtering, collaborative filtering (CF) and demographic filtering [3, 12]. Among these techniques, CF is the most widely used and prevalent technique [12]. Collaborative filtering (CF) recommends items to active users based on those users who have similar tastes

in the past. When a user has rated a few items, a reliable recommendation is not possible for that user. This problem is termed as a cold-start user problem. Furthermore, traditional CF also suffers from the sparsity problem [12].

The growing popularity of open social network and trend to integrate e-commerce applications with RS have generated an increased interest toward developing trust aware RS as people rely more on those recommendations suggested by trustworthy people in real life [7]. In these trust aware RS, usually a trust network is used to search more likely neighbors by establishing a relationship between users that are not sharing any co-rated items. Trust-aware CF approaches can be broadly classified into two categories: namely, explicit trust model [1, 4, 5] or implicit trust model [2, 6, 7, 10]. Recently, a lot of work has been carried out by elicitation of trust values into collaborative RSs for improving the accuracy of predictions and handling the sparsity as well as cold start problems. In contrast to other trust-aware recommendation methods, our approach also exploits distrust links among users. The effect of distrust has not been much analyzed in the realm of RS due to the absence of available data sets representing both the trust and the distrust values for a particular person [8]. Our work in this paper is an attempt toward developing trust-distrust enhanced recommendations model based on the novel similarity measure that combines user ratings and trust values for generating more quality recommendations. Our work has the following main research contributions:

- Designing a novel similarity measure for CF based on the computed trust values between users.
- Handling the problems of new user and sparsity by utilizing propagation operator based on trust-distrust values.
- Comparative analysis of proposed recommendation strategies using of trust-distrust models.

The rest of this paper is organized as follows: Sect. 2 covers related work. Section 3 describes the overall framework of our approach. Computational experiments and results are given in Sect. 4. Finally, we conclude our work in Sect. 5.

2 Related Work

Collaborative filtering and explanation of direct and indirect models of Trust and Distrust are described in this section.

2.1 Collaborative Filtering

Collaborative filtering, follows the principle of ‘word of mouth’ where similar users provide suggestions to users. The following three steps are required to generate recommendations to users in CF based RS.

- Step 1 (Similarity Computation): It computes the similarity between active users (u_a) and other user (u) by using various similarity measures such as cosine similarity, Pearson correlation, jaccard similarity. The most widely used similarity measure in CF is Pearson similarity measure which is defined below:

$$Sim(u_a, u) = \frac{\sum_{i \in I} (r_{u_a, i} - \bar{r}_{u_a})(r_{u, i} - \bar{r}_u)}{\sqrt{\sum_{i \in I} (r_{u_a, i} - \bar{r}_{u_a})^2} \sqrt{\sum_{i \in I} (r_{u, i} - \bar{r}_u)^2}} \quad (1)$$

where, $r_{u_a, i}$ - Rating provided by user u_a on item i

\bar{r}_u - Mean rating of user u

I - Set of corated items.

- Step 2 (Neighbourhood set formation): Usually top k similar users are selected in the neighbourhood sets. Alternatively the neighbourhood set can be generated through predefined similarity threshold.
- Step 3 (Prediction and Recommendation): It predicts an unknown rating of a target item for an active user based on the neighbourhood set using following formula:

$$P_{u_a, m} = \bar{r}_{u_a} + \frac{\sum_{u \in N(u_a)} Sim(u_a, u)(r_{u, m} - \bar{r}_{u_a})}{\sum_{u \in N(u_a)} Sim(u_a, u)} \quad (2)$$

where, $N(u_a)$ - Set of neighbours to user u_a

$P_{u_a, m}$ - Represents the predictive rating of active user u_a on item m

$r_{u, m}$ is the rating of user u who is a neighbour of user u_a . Finally highly predicted items will be recommended to active users.

However, similarity based CF suffers several problems such as, cold-start and sparsity that could affect the precision of recommendations [3, 12]. To generate effective recommendations by dealing with these concerns, many studies have been conducted by eliciting trust values into collaborative recommender system. In these studies, a trust network is built between users that may be helpful to RS [4–6]. It is also indicated that a user is much more confident on trusted user rather than a stranger. Since this trusted user may also trust his friend’s opinion in recursive manner by propagating trust values. Guha et al. [2] was the first one who utilized the idea of transitivity of trust and developed a framework for trust propagation. In the area of RS, a new trend about distrust is also investigated recently. Victor et al. [8] developed trust assessment scheme between unconnected pairs in a trust and distrust network by using trust and distrust propagation and aggregation operators and explored various ways in which distrust information can be utilized in a fine-tuned network using the Epinion data set. Since this data set does not include assignment of pair (trust, distrust) to individuals, the propagation/aggregation operators have not been fully analyzed especially in inconsistent situations [8, 9].

2.2 Trust Model

Trust models can be classified into two categories, namely explicit trust model and implicit trust model. An explicit trust model deals with direct linking

between users where users specify their trust values to directly connected users [1, 4, 5]. However, implicit trust model computes trust values among users either by propagating trust values or computing trust values based on available ratings on items [6, 7] (Table 1).

Table 1. Trust model

Trust	Trust and distrust
Lathia et al. [6] (implicit trust)	Kant et al. [11] (implicit trust)
Bharadwaj et al. [7] (implicit trust)	Guha et al. [2] (implicit trust)
Golbeck [4] (explicit trust)	Victor et al. [8] (implicit trust)
Massa et al. [1] (explicit trust)	

3 Trust Distrust Enhanced Recommendation Framework

In this section, we will discuss about our proposed trust-distrust enhanced recommendation framework. For a RS, let $U = \{u_1, u_2, u_3, \dots, u_n\}$ be the set of n users and $I = \{i_1, i_2, i_3, \dots, i_m\}$ is the set of m items in the system. Each user u_i rated a set of items and rating of u_i on i_j is expressed as r_{u_i, i_j} . Our proposed system has following three phases which are depicted in Fig. 1. The details about these phases are given below:

Phase 1. (Effective Similarity Computation based on trust values): We have computed effective similarity through three steps which are discussed below:

- Step 1 (Similarity computation): We have computed the similarity between active user u_a and a user u by using Eq. 1.
- Step 2 (Trust-Distrust Computation): We have evaluated trust and distrust values between active user u_a on user u by using following equations:

$$Trust_{u_a}(u) = \frac{2 * rec_{trust} * exp_{trust}(u_a, u)}{rec_{trust} + exp_{trust}(u_a, u)} \tag{3}$$

where, rec_{trust} and exp_{trust} will be computed by utilizing the computational models [8, 11]

$$Dis_{u_a}(u) = \frac{2 * rec_{dis} * exp_{dis}(u_a, u)}{rec_{dis} + exp_{dis}(u_a, u)} \tag{4}$$

where, rec_{dis} and exp_{dis} will be computed by utilizing the computational models [11].

- Step 3 (Effective Similarity): In real life, users are more confident on those users who are more trustworthy. Therefore, we have embedded similarity with trust value to compute effective similarity measure $Sim'(u_a, u)$ between active user u_a and a user u by using following formula:

$$Sim'(u_a, u) = \frac{(w_1 * Sim(u_a, u)) + (w_2 * Trust(u_a, u))}{w_1 + w_2} \tag{5}$$

The reason for fusing these two types of information is based on the observation that the similarity and social trust among users may not be highly correlated.

Here, weights are decided experimentally and these values (w_1 and w_2) are normalized in the range of $[0,1]$.

Phase 2. (Neighbourhood set construction based on distrust as a filter): At this stage, the distrust is used as a means to filter out neighbours before the recommendations so that only the most trusted neighbours can participate in the recommendation process. Thus, the distrust system will be implemented on the neighbourhood set to filtered out most distrust user from neighbourhood set.

Phase 3. (Prediction and Recommendations): The selected neighbourhood set after phase 2 is used to predict the ratings of all unseen items for an active user using Eq. 2. Finally top predicted items can be recommended to the active user.

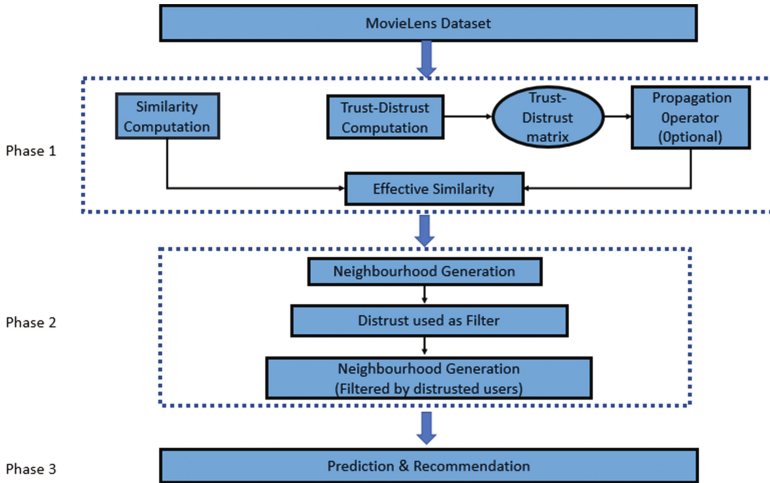


Fig. 1. Three phases of our proposed recommendation framework

4 Experiment Setup

To show the effectiveness of our proposed approaches we conducted several experiments on MovieLens dataset.

4.1 Design of Experiments

MovieLens data set contains 100,000 ratings provided by 943 users on 1682 movies on a using 5 point rating scale [11]. We divided the whole MovieLens

dataset into 5 splits. Each split contains 200 users. For each split, we selected 50 active users randomly and the remaining 150 users are considered as training users in each split. Further, we divided ratings of each active user into two sets namely training movies [60%] and test movies [40%]. Training movies are used for constructing neighbourhood generation and trust-distrust computation. We repeated all experiments on each split five times in order to reduce the inherent bias if it exists. In all experiments we kept fixed neighbourhood size (k) which is decided by verifying different values of k in the experiments.

4.2 Performance Evaluation

We have used following performance measures for the evaluation of our proposed approaches

- **Mean absolute error (MAE):** MAE represents the difference between actual ratings and predicted ratings.

$$MAE = \frac{1}{n} \sum_{i=1}^n |a_i - p_i| \quad (6)$$

where, a_i is actual rating.

p_i is predicted rating.

n is total no of predicted item.

- **Precision:** Precision, measuring correctness of recommendation, is defined as the ratio of the number of selected items to the number of recommended items.

$$precision = \frac{\text{Number of item recommended}}{\text{Total number of recommended item}} \quad (7)$$

- **Recall:** Recall is a measure of completeness. It determines the ratio of good items retrieved to all good items. In other words, it computes the fraction all good movies recommended.

$$recall = \frac{|\text{good movies recommended}|}{|\text{all good movies}|} \quad (8)$$

- **F-measure:** The f-measure is the harmonic mean of precision and recall

$$f\text{-measure} = 2 \times \frac{precision \times recall}{precision + recall} \quad (9)$$

- **Percentage of correct prediction PCP:** PCP is defined as the ratio of Correctly predicted items to the number of rated items.

$$PCP = \frac{\text{Correctly predicted item}}{\text{Total number of rated item}} * 100 \quad (10)$$

4.3 Experiments

We have compared our approaches namely Trust Distrust Pearson Collaborative Filtering (TD_PCF), Trust Pearson Collaborative Filtering with propagation (TPCF_PROP) and Trust distrust Pearson Collaborative Filtering with Propagation (TD_PCF_PROP) with the following approaches such as:

- Pearson Collaborative Filtering (PCF) [15]
- Trust based Collaborative Filtering (TCF) [1]
- Trust Distrust Collaborative Filtering (TDCF) [9]
- Trust Collaborative Filtering with propagation (TCF_PROP)
- Trust distrust Collaborative Filtering with Propagation (TD_CF_PROP) [2]
- Trust Based Weight Collaborative Filtering (TBW) [4,9]
- Trust Based Filtering Collaborative Filtering (TBF) [9]
- Ensemble Trust Collaborative Filtering (ETCF) [16].

4.4 Result

To demonstrate the effectiveness of the proposed approaches TD_PCF_PRO, TPCF_PRO and TDPCF, we analyzed the results for the MAE, PCP, precision and f-measure as shown in Tables 2, 3, 4 and 5. In these tables, last row indicates the average performance over five splits. The lower values of MAE implies the better performance of the approach. Similarly, higher values of PCP, precision and f-measure also indicate the better performance. Based on these tables, we

Table 2. Performance comparison on various approaches on MAE

SPLIT	PCF	TCF	TCF PROP	TPCF PROP	TDCF	TDPCF	TD_CF PROP	TD_PCF PROP	TBW	TBF	ETCF
Split1	0.841	0.826	0.821	0.841	0.864	0.842	0.821	0.837	2.761	0.833	3.124
Split2	0.836	0.827	0.799	0.822	0.954	0.835	0.799	0.822	2.767	0.824	3.023
Split3	0.864	0.861	0.842	0.825	0.940	0.865	0.845	0.826	2.831	0.852	3.013
Split4	0.869	0.863	0.846	0.827	0.988	0.867	0.845	0.820	2.931	0.863	3.155
Split5	0.962	0.932	0.905	0.847	1.125	0.748	0.905	0.839	2.899	0.957	2.951
MEAN	0.874	0.862	0.843	0.833	0.975	0.832	0.844	0.829	2.838	0.866	3.053

Table 3. Performance comparison on various approaches on PCP

SPLIT	PCF	TCF	TCF PROP	TPCF PROP	TDCF	TDPCF	TD_CF PROP	TD_PCF PROP	TBW	TBF	ETCF
Split1	35.75	36.56	37.37	37.54	36.17	35.69	37.37	37.49	6.46	36.25	3.00
Split2	33.87	34.56	38.30	39.42	35.17	33.79	38.30	38.99	5.11	34.46	2.15
Split3	35.39	35.84	39.05	37.57	36.63	35.48	38.90	37.12	6.14	36.23	3.84
Split4	37.74	38.02	40.35	39.78	37.79	37.75	40.45	39.77	7.79	37.76	2.99
Split5	30.95	31.75	35.73	39.59	31.26	30.61	35.73	41.24	7.40	31.37	4.12
MEAN	34.74	35.35	38.16	38.78	35.41	34.67	38.1490	38.93	6.58	35.21	3.22

Table 4. Performance comparison on various approaches on Precision

SPLIT	PCF	TCF	TCF PROP	TPCF PROP	TDCF	TDPCF	TD_CF PROP	TD_PCF PROP	TBW	TBF	ETCF
Split1	0.836	0.820	0.802	0.888	0.811	0.837	0.802	0.888	0.018	0.833	0.003
Split2	0.863	0.862	0.861	0.881	0.830	0.865	0.860	0.895	0.009	0.859	0.005
Split3	0.973	0.947	0.947	0.916	0.932	0.972	0.947	0.905	0.007	0.965	0.011
Split4	0.844	0.868	0.875	0.894	0.812	0.845	0.875	0.903	0.013	0.846	0.012
Split5	0.845	0.854	0.859	0.884	0.796	0.844	0.859	0.886	0.015	0.839	0.006
MEAN	0.872	0.870	0.869	0.893	0.836	0.873	0.869	0.895	0.012	0.869	0.007

Table 5. Performance comparison on various approaches on F-Measure

SPLIT	PCF	TCF	TCF PROP	TPCF PROP	TDCF	TDPCF	TD_CF PROP	TD_PCF PROP	TBW	TBF	ETCF
Split1	0.767	0.768	0.787	0.835	0.767	0.767	0.787	0.836	0.032	0.768	0.006
Split2	0.813	0.817	0.829	0.832	0.789	0.816	0.829	0.842	0.016	0.811	0.009
Split3	0.904	0.891	0.896	0.859	0.876	0.904	0.896	0.854	0.013	0.899	0.019
Split4	0.795	0.829	0.850	0.847	0.780	0.797	0.851	0.855	0.022	0.804	0.022
Split5	0.778	0.792	0.809	0.829	0.737	0.777	0.809	0.831	0.028	0.776	0.011
MEAN	0.811	0.819	0.834	0.840	0.789	0.812	0.834	0.843	0.022	0.812	0.013

can say that our proposed approaches namely, TD_PCF_PRO, TPCF_PRO and TDPCF, outperform other approaches in terms of various performance evaluation schemes.

5 Conclusion

Recommender systems are one of the recent invention for dealing with information overload problem by identifying more relevant items to users based on their preferences. Collaborative filtering is the most successful recommendation technique in the area of RS. However, the new user and sparsity are major concerns. In this work, we have proposed trust distrust enhanced recommendation framework where effective similarity is suggested for using the utility of trust and similarity factor in the construction of neighbourhood set. For more efficient neighbours, we have filtered out the distrusted user from the neighbourhood set. Further, we have investigated the use of trust distrust based propagation operator in resolving the new user and sparsity problems. Finally, experimental results demonstrated that our proposed strategy were superior to traditional collaborative filtering and other existing trust aware recommendation strategies.

References

1. Massa, P., Avesani, P.: Trust-aware collaborative filtering for recommender systems. In: Meersman, R., Tari, Z. (eds.) OTM 2004. LNCS, vol. 3290, pp. 492–508. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30468-5_31

2. Guha, R., et al.: Propagation of trust and distrust. In: Proceedings of the 13th International Conference on World Wide Web. ACM (2004)
3. Adamavicius, G., Tuzhilin, A.: Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **17**(6), 734–749 (2005)
4. Golbeck, J.A.: Computing and applying trust in web-based social networks. Dissertation (2005)
5. O'Donovan, J., Smyth, B.: Trust in recommender systems. In: Proceedings of the 10th International Conference on Intelligent User Interfaces. ACM (2005)
6. Lathia, N., Hailes, S., Capra, L.: Trust-based collaborative filtering. In: Karabulut, Y., Mitchell, J., Herrmann, P., Jensen, C.D. (eds.) IFIPTM 2008. IFIP-TIFIP, vol. 263, pp. 119–134. Springer, Boston (2008). https://doi.org/10.1007/978-0-387-09428-1_8
7. Bharadwaj, K.K., Al-Shamri, M.Y.H.: Fuzzy computational models for trust and reputation systems. *Electron. Commer. Res. Appl.* **8**(1), 37–47 (2009)
8. Victor, P., et al.: Gradual trust and distrust in recommender systems. *Fuzzy Sets Syst.* **160**(10), 1367–1382 (2009)
9. Victor, P., et al.: Trust-and distrust-based recommendations for controversial reviews. In: Web Science Conference (WebSci 2009: Society On-Line). No. 161 (2009)
10. Victor, P., et al.: Practical aggregation operators for gradual trust and distrust. *Fuzzy Sets Syst.* **184**(1), 126–147 (2011)
11. Kant, V., Bharadwaj, K.K.: Fuzzy computational models of trust and distrust for enhanced recommendations. *Int. J. Intell. Syst.* **28**(4), 332–365 (2013)
12. Bobadilla, J., et al.: Recommender systems survey. *Knowl.-Based Syst.* **46**, 109–132 (2013)
13. Anand, D., Bharadwaj, K.K.: Pruning trust-distrust network via reliability and risk estimates for quality recommendations. *Social Network Anal. Min.* **3**(1), 65–84 (2013)
14. Lee, W.-P., Ma, C.-Y.: Enhancing collaborative recommendation performance by combining user preference and trust-distrust propagation in social networks. *Knowl.-Based Syst.* **106**, 125–134 (2016)
15. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) *The Adaptive Web*. LNCS, vol. 4321, pp. 291–324. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72079-9_9
16. Victor, P., et al.: A comparative analysis of trust-enhanced recommenders for controversial items. In: ICWSM (2009)

Language Identification Based on the Variations in Intonation Using Multi-classifier Systems

Shinjini Ghosh 

South Point High School, 82/7A Ballygunge Place, Kolkata 700019, India
annabelle.rosie00@gmail.com

Abstract. In this article we make use of the characteristics of tonal languages and machine learning methodologies to understand the patterns in them. Instead of analyzing the absolute pitch or frequency, we analyze how one tone transitions to another in speech. Features (namely, zero crossing count, short time energy, minimum formant frequency, maximum formant frequency) are extracted using the tonal transitions over segments of audio signals. We have developed a multi-classifier system using four classifiers, namely maximum likelihood estimate (MLE), minimum distance classifier (MDC), k-nearest neighbor (kNN) classifier and fuzzy k-NN classifier to automatically identify tonal languages from audio signals. Initially, each individual classifier is trained with existing known data represented by the extracted features. The trained classifier is then used for language identification. Results obtained from these classifiers are combined to generate the final output. Experiments are conducted using three different tonal languages, namely, Chinese, Thai and Vietnamese. The output reveals that the developed multi-classifier model is able to produce promising results. The extracted features produced better results in comparison to usually used frequency value (as a feature). Ensemble of classifiers is a better tool than using individual classifiers.

Keywords: Tonal language · Language identification · Classification
Multi-classifier

1 Introduction

Speaking plays an indispensable role in communication. We utilize our vocal apparatus to speak through various articulatory processes including variations in the manner, place and the intonation or frequency. Apart from the inventory of sounds, we also extensively use variations in the tone to convey additional meaning(s). Communicating in tonal language requires one to follow the usual tonal transitions of that language, as straying from that changes the entire meaning of words and sentences. Transition of tone, *intonation*, refers to temporal changes of fundamental frequency and it is a key feature of a tonal language. A tonal language can be identified by modeling its intonation (see Fig. 1). Various features could be extracted from the audio signals to model these tonal transitions. As machine learning techniques try to emulate the human learning system, it was thought to use such methodologies to design and build an automated system as it

could work in a more natural way. In this paper the characteristics of tonal languages and machine learning methodologies are used to understand the patterns in such languages.



Fig. 1. Example of tonal variations in Chinese

Language identification primarily involves understanding and determination of four key features: Acoustic Phonetics, Prosodics, Phonotactics, and Vocabulary [1]. Of these, prosodics deals with rhythm, syllabification and intonation. Major research on it is concentrated on using Gaussian Mixture Model, Hidden Markov Model, Multilayer Perceptron and Support Vector Machine based techniques [2–5]. An extensive review of previous work on automatic Language Identification (LID) can be found in [2–4]. Itahashi et al., in 1994, developed a spoken language discrimination method based on parameters derived from fundamental frequency contours of speech and suggested that prosodic feature could be a base of language identification [6]. In 2006 A ‘bag-of-sounds’ model for LID was proposed by Tong et al. using Gaussian modeling techniques to tokenize sound into language specific phonemes [7]. Fusion of five features (spectrum, duration, pitch, n-gram phonotactic, bag-of-sounds) at different levels of abstraction were studied. In 2009, Rao and Yegnanarayana modeled the intonation for Indian languages in terms of variation of frequency about fundamental frequency using feed-forward neural networks (FFNNs) [8]. Visual features were used for language identification by Newman and Cox [9]. In [10] the authors transformed the utterances to a low dimensional i-vector representation upon which language classification is done. Yencken’s *A Great Language Game* in 2013 [11] inspired us to carry out this work.

In this article, as compared to earlier works, features specifically representing tonal languages are extracted over segments of audio signals. As mentioned earlier, the extracted features include zero crossing count, short time energy, minimum formant frequency, and maximum formant frequency which better describe the tonal transitions over segments of audio signals. Thereafter, four different classifiers [12] are used, both in isolation as well as in combination, to identify the language using these tonal features. Use of some of these features and an ensemble of classifiers is a new way to identify tonal languages. Results found are quite promising in comparison to the usual frequency based single classifier systems.

The rest of the article is organized as follows. Section 2 describes the proposed methodology for tonal language identification. In Sect. 3 implementation details are described. Results and discussion are presented in Sect. 4. Finally, conclusion and future scope are provided in Sect. 5.

2 Proposed Work

The aim of the present work is to automatically categorize different tonal languages using machine learning techniques. As mentioned earlier, in the present work, from each of the audio signals several features are extracted. Besides frequency, four different features, namely, zero crossing count, short time energy, minimum formant frequency, and maximum formant frequency are generated using the tonal transitions over frames of audio signals. Four classifiers, namely, maximum likelihood estimate (MLE), minimum distance classifier (MDC), k-nearest neighbour (kNN) classifier and fuzzy k-NN classifier [12, 13] are considered to automatically identify tonal languages from the audio signals. For MDC, k-NN and fuzzy k-NN, Euclidean distance measure is used. Initially, each individual classifier is trained with existing known audio data represented by the extracted features. The trained classifier is then used for language identification. A multi-classifier system is developed that takes the output of these classifiers as input and combines them using majority voting principle to arrive at the final decision. It is well established in the literature that an ensemble of classifiers usually produces better results in contrast to single classifiers. This is why we have opted for a combination of classifiers. Also, we have used different extracted features rather than only the frequency values to represent variations in tonal languages.

Block diagram of the proposed multi-classifier based system for tonal language identification is shown in Fig. 2. The step-by-step procedures of each of the classifier based identification models are described and their corresponding block diagrams are also given in Figs. 3, 4, 5 and 6.

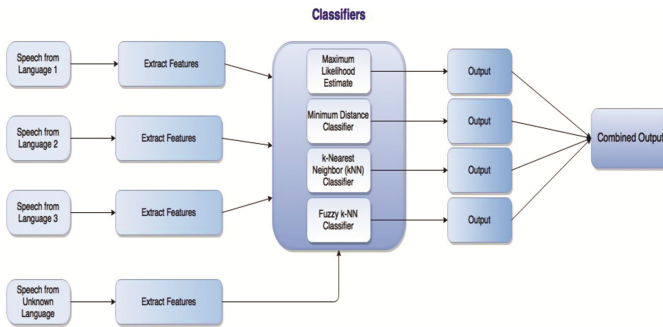


Fig. 2. Block diagram of the proposed model

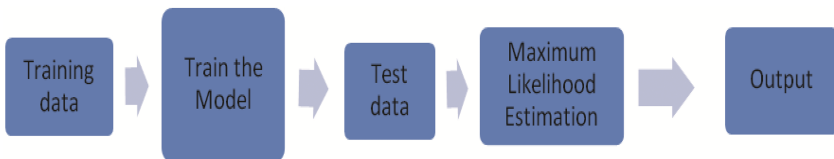


Fig. 3. Block diagram of maximum likelihood estimate

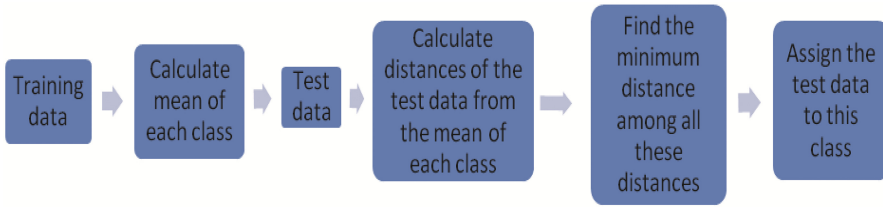


Fig. 4. Block diagram of minimum distance classifier

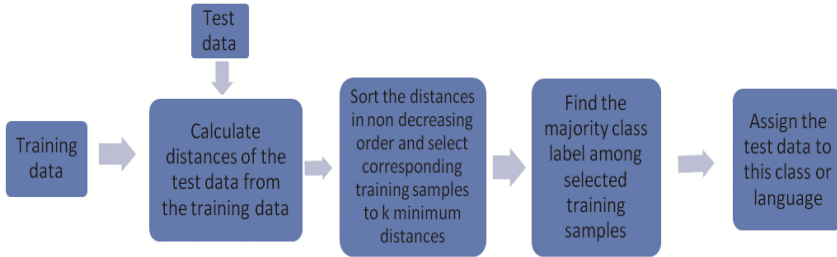


Fig. 5. Block diagram of k-nearest neighbor classifier

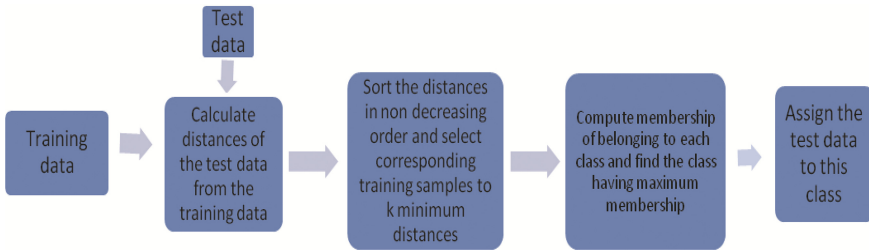


Fig. 6. Block diagram of fuzzy k-nearest neighbor classifier

Algorithm 1 (Maximum Likelihood Estimation (MLE))

- (a) Procure audio data (raw data) from various tonal languages.
- (b) Convert them to MIDI format. Extract frequency values (or, features) from each MIDI file using First Fourier Transform (FFT) [14].
- (c) Extract zero crossing count, short time energy, minimum formant frequency and maximum formant frequency from each audio. These form our Corpus for training. Similarly, test dataset is generated.
- (d) Estimate the likelihood of the test data from the training dataset of each of the languages.
- (e) Sort the likelihood values in descending order.
- (f) Categorize the test data into that language with which it has the maximum likelihood.

Algorithm 2 (Minimum Distance Classifier (MDC))

- (a) First three steps are identical to Algorithm 1.*
- (b) Train the minimum distance classifier (i.e., calculate the mean of each class) using the Corpus (training dataset).*
- (c) Calculate distances of the test data from the mean of all classes.*
- (d) Label the test data with the class label (language) whose mean is at the minimum distance.*

Algorithm 3 (k-Nearest Neighbor Classifier (kNN))

- (a) First three steps are identical to Algorithm 1.*
- (b) Calculate distances of the test data from all the training data.*
- (c) Sort the distances in non-decreasing order and select the training samples (with their class labels) corresponding to the first k (a positive integer) minimum distances.*
- (d) Categorize the test data into that language from which majority of these k training data comes.*

Algorithm 4 (Fuzzy k-Nearest Neighbor Classifier (Fuzzy kNN))

- (a) First three steps are identical to Algorithm 3.*
- (b) Assign class membership to the selected training data (inversely proportional to these distances).*
- (c) Add up the membership values of all the patterns (out of k) coming from each class.*
- (d) Find the maximum membership value.*
- (e) Categorize the test data into the language for which this membership value is the maximum.*

3 Implementation

As mentioned earlier, in the present work, from each of the audio signals, several features are extracted, which are then used to classify tonal languages using various classifiers. Experiments are carried out with an Intel Core i3 machine with 2.4 GHz CPU. Datasets used for the experiment and the method of feature extraction are described in the subsequent sections.

A. Datasets Used

Data from three different tonal languages namely, Chinese, Thai and Vietnamese are collected for experimentation. Our repository consists of 26 Chinese, 119 Thai and 23 Vietnamese audio samples each of 2 s duration, consisting of 191 frames with no time gap in between them. Out of this, 10 audio samples from each class have been utilized for training and the rest for testing. Multiple test data sets have been collected. Datasets have been obtained from various places over the Internet and from friends and acquaintances. Thai data and some Chinese data are obtained from School of Information Technology, King Mongkut's University of Technology, Thonburi, Thailand. Time versus frequency plots for a fragment of sample speech of each of the three languages (generated using Sonic Visualizer [15]) are shown in Figs. 7, 8 and 9.

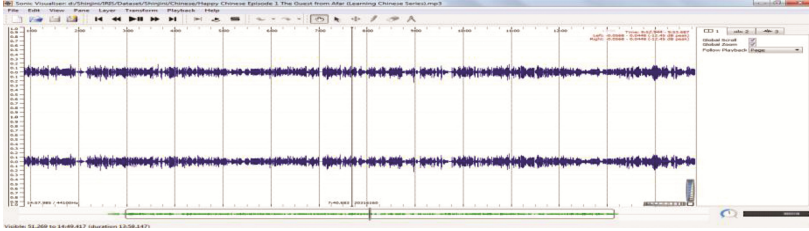


Fig. 7. Time vs. Frequency plot for Chinese data

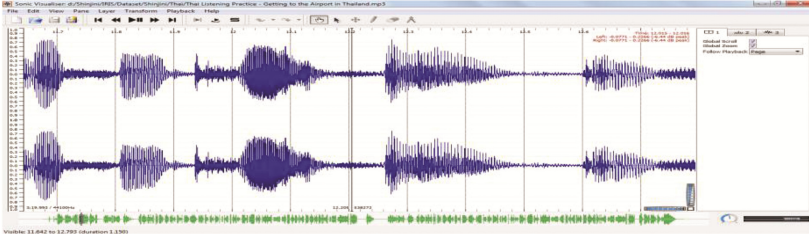


Fig. 8. Time vs. Frequency plot for Thai data

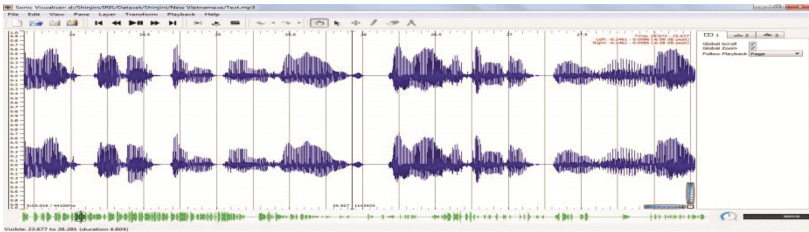


Fig. 9. Time vs. Frequency plot for Vietnamese data

B. Feature Extraction

Collected audio data (e.g., in MP3 format) is converted into corresponding MIDI file using MIDI conversion software widely available over the net. Python package is used to extract frequency values using FFT from these MIDI files. MATLAB is used to extract features like zero crossing count, short time energy, minimum formant frequency and maximum formant frequency. For training, testing and generating results, the algorithms are implemented using C Programming language. To preprocess data, background noise is removed with an open source software named Audacity.

4 Results and Discussion

To show the effectiveness of the proposed model, experiments are conducted with three different tonal languages (Chinese, Thai and Vietnamese). Each feature vector has frequency and a combination of zero crossing count, short time energy, and minimum

& maximum formants frequencies as its component. Investigation has also been carried out with preprocessed audio data. For all the cases, results (shown in Table 1) are obtained considering the classifiers in isolation and in combination. For each of the cases, class-wise accuracy and overall accuracy are noted. The best performance is marked in bold.

Table 1. Performance of the proposed multi-classifier system for tonal language identification

	Languages used	Classifiers used				
		MLE	MDC	kNN	FkNN	Combination of classifiers
Percentage of accuracy (feature: frequency)	Chinese	16.67	66.67	83.33	83.33	83.33
	Thai	23.85	41.28	14.68	22.94	26.61
	Vietnamese	100.0	61.54	53.85	53.85	61.54
	Overall	31.25	44.53	21.88	28.91	32.81
Percentage of accuracy (feature: zero crossing count, short time energy, minimum and maximum formant frequencies)	Chinese	62.50	87.50	100.0	100.0	100.0
	Thai	40.54	90.09	97.30	95.50	94.59
	Vietnamese	100.0	100.0	100.0	20.0	100.0
	Overall	44.35	90.32	97.58	92.74	95.16
Percentage of accuracy (preprocessed data) (feature: zero crossing count, short time energy, minimum and maximum formant frequencies)	Chinese	100.0	100.0	100.0	100.0	100.0
	Thai	88.07	88.99	88.99	88.99	88.99
	Vietnamese	100.0	100.0	100.0	100.0	100.0
	Overall	89.92	90.70	90.70	90.70	90.70

It is seen from Table 1 that the classification results are promising in nature. It is observed that a combination of the four features (zero crossing count, short time energy, and minimum & maximum formants frequencies) as feature vector produces better results as compared to only frequency (the popularly used case). Results obtained using MDC, kNN, and fuzzy k-NN are better than those obtained using MLE. As expected, for preprocessed data, the performance of MLE is significantly improved. Moreover, for some datasets, fuzzy k-NN produced better results as compared to other classifiers. It is seen that combination of classifiers showed better performance than individual classifiers in many cases. As the features are peculiar for tonal languages, they can be used to distinguish tonal and non-tonal languages also.

5 Conclusion

Four classification models namely, maximum likelihood estimate, minimum distance classifier, k-nearest neighbor classifier and fuzzy k-nearest neighbor classifier are initially developed to automatically identify tonal languages. Results obtained from these classifiers are combined to have a final outcome. Besides frequency values, four other features (zero crossing count, short time energy, minimum formant frequency, maximum formant frequency) are extracted over segmented audio input signals to represent them. Experiments are conducted with three tonal languages namely, Chinese, Thai, and Vietnamese. From the results it is found that intonation provides a key insight into the characteristic identity of such languages.

To study the robustness of the model, investigation needs to be carried out for finding out the superiority of classification approaches using a wide variety of languages (tonal and non-tonal) embedding several other features. Applicability of other classification methodologies (e.g., neural networks, SVM) will also be explored. The proposed technique can also be used to distinguish tonal and non-tonal languages.

Acknowledgment. An earlier version of this work has been presented at the Intel International Science and Engineering Fair (Intel ISEF), held at Los Angeles, USA in May 2017 and won a Grand Award. The author would like to acknowledge her School teacher, Dr. Partha Pratim Roy, for advising her throughout the course of this work. Thanks are due to the Intel Initiative for Research and Innovation in Science (IRIS) Scientific Review Committee and her mentors, for their valuable comments. The author also acknowledges Rahul Roy and Ajoy Mondal, her parents' students, for helping her in conducting the experiments.

References

1. Jurafsky, D., Martin, J.H.: *Speech and Language Processing*, 2nd edn. Pearson, New Delhi (2014)
2. Muthusamy, Y.K., Barnard, E., Cole, R.A.: Reviewing automatic language identification. *IEEE Sign. Process. Mag.* **11**, 33–41 (1994)
3. Zissman, M.A.: Automatic language identification of telephone speech. *Lincoln Laboratory Manual*, MIT, USA, vol. 8, no. 2, pp. 115–144 (1995)
4. Ambikairajah, E., Li, H., Wang, L., Yin, B., Sethu, V.: Language identification: a tutorial. *IEEE Circ. Syst. Mag.* **11**(2), 82–108 (2011)
5. Ng, R.W.M., Lee, T., Leung, C., Ma, B., Li, H.: Spoken language recognition with prosodic features. *IEEE Trans. Audio Speech Lang. Process.* **21**(9), 1841–1852 (2013)
6. Itahashi, S., Zhou, J.X., Tanaka, K.: Spoken language discrimination using speech fundamental frequency. In: *Proceedings of Third International Conference on Spoken Language Processing*, Japan, vol. 4, pp. 1899–1902 (1994)
7. Tong, R., Ma, B., Zhu, D., Li, H., Chng, E.S.: Integrating acoustic, prosodic and phonotactic features for spoken language identification. In: *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. I 205–I 208 (2006)
8. Rao, K.S., Yegnanarayana, B.: Intonation modeling for Indian languages. *J. Comput. Speech Lang.* **23**, 240–256 (2009)
9. Newman, J.L., Cox, S.J.: Language identification using visual features. *IEEE Trans. Audio Speech Lang. Process.* **20**(7), 1936–1947 (2012)
10. Segbroeck, M., Travadi, R., Narayanan, S.S.: Rapid language identification. *IEEE Trans. Audio Speech Lang. Process.* **23**(7), 1118–1129 (2015)
11. Yencken, L.: *The great language game* (2013). www.greatlanguagegame.com
12. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*, 2nd edn. Wiley, New York (2001)
13. Theodoridis, S., Koutroumbas, K.: *Pattern Recognition*, 4th edn. Elsevier, New York (2008)
14. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*, 3rd edn. Pearson Education, New Delhi (2009)
15. Cannam, C., Landone, C., Sandler, M.: Sonic visualiser: an open source application for viewing, analysing, and annotating music audio files. In: *Proceedings of the 18th ACM International Conference on Multimedia*, pp. 1467–1468 (2010)

Cognitive Decision Making for Navigation Assistance Based on Intent Recognition

Sumant Pushp¹(✉), Basant Bhardwaj¹, and Shyamanta M. Hazarika²

¹ National Institute of Technology, New Delhi 110040, India
sumantpushp@gmail.com

² Indian Institute of Technology, Guwahati 781039, Assam, India

Abstract. Within rehabilitation robotics, machines are being designed to help human in activities of everyday life. Mobility is an essential component for independent living. Autonomous machines with their high degree of mobility are becoming an integral part of assistive devices leading to a number of developments in mobility assistance. This is primarily in terms of smart wheelchairs embodied with agents. Autonomous agents keep an eye on irregularities during navigation and trigger corrections whenever required. They behave as teammates for the human wheelchair user. Such agents will be more effective if it's behavior is closer to human or it is intelligent enough to understand the possible course of action taken by the human user. Therefore recognizing intention of the human driver and surrounding vehicles is an essential task. We have formulated a fuzzy model for the prediction of intention. A qualitative distance and orientation mechanism have been adopted, where few environment features are taken to show how the prediction of intention can improve the ability of decision making.

Keywords: Intent recognition · Autonomous vehicle navigation
Motion planning · Obstacle avoidance

1 Introduction

Autonomous decision systems in outdoor navigation represent a convergence of diverse areas of research. The central objective is to effectively work in a real world environment that has not been specifically engineered. The evolution of such system is challenging. For a truly autonomous robot, systems designed with a preformed sequence of operations within a highly constrained environment are not acceptable. Such robotic systems usually fail to work in an unexplored scenario. Many methods have been proposed for robot navigation. A very basic inertial navigation method which provide dynamic information through direct measurements was proposed in 1995 [3]. The system calculates distance at real time and avoid collisions. A force based potential field navigation method was proposed [1]. Here obstacles exert repulsive forces onto the robot, while the target applies an attractive force. The resultant force determines the subsequent

direction and speed of travel. Vector field histogram [4], Robust Monte Carlo Navigation [19], occupancy grids [7], map matching, and many others techniques have been used for navigation. Despite recent advances in autonomous robots, a number of difficulties need to be sorted to achieve a true autonomous system. The wide variety of uncertainty arising out of an unstructured environment is a major barrier for such systems. We require a methodology which can provide the probabilistic future events within its immediate environment and take a deliberate decision.

We believe that adding cognitive reasoning into intelligent systems can lead to more natural and human compatible behavior of the resulting system [9]. Recognition of ‘intent’ of the teammates or other agents in the vicinity is one such cognitive ability. Intent recognition involves prediction of intentions of an agent, usually by observing an agent or a group of agents [12] in a dynamic environment. It is a proactive approach for decision making [17] and have been successfully used in service robots designed for assistance [2, 10, 16]. In this paper, a fuzzy model for prediction of intention is presented. Under the assumption of availability of few environmental features, we apply a fuzzy based prediction of turning behavior of surrounding vehicles in order to find a safe and smooth path for the subject vehicle. Combination of two qualitative spatial reasoning methods [5] are incorporated to deal with the distance and orientation.

2 Model of Intention Prediction

Any approach used to control dynamic system needs to use some knowledge or model of the system to be controlled. The kinematics and dynamics of a subject vehicle may be complex and nonlinear [8]. Further, the interaction between the surrounding vehicles is hard to model in general. This motivated several communities to use fuzzy control techniques [11, 13, 15]. We have designed an intention based decision system to model the behavior of an autonomous mobile agent (Fig. 1).

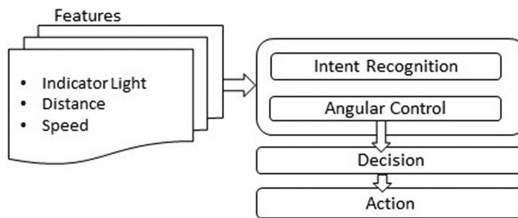


Fig. 1. Model of intention based system

The Model Consists of few predetermined features which are available to the system and a fuzzy module to predict intention of motion of surrounding vehicles. The angular control module finds a safe direction and angle of the

subject vehicle exploiting qualitative orientation $OPRA_m$ [14] and an absolute distance calculus [6].

2.1 Features

Three features related to surrounding vehicles are analyzed, which is available to the system as input parameters. Two quantitative features *distance* and *velocity* and a qualitative “Indicator” signal is considered to model the behaviour of system. Though only these number of features are not sufficient to justify a robust system however, taking few provide simplicity. And suffice for a first step towards establishing a claim that intention based approaches have a significant impact on cognitive decision making.

We consider the indicator signal of surrounding vehicle as one of the features to understand the qualitative intention of surrounding vehicles. Usually an indicator signal is mounted as a uni-colour light on both end of vehicles. We assume F_I which denote the indicator feature, which may have status ON or OFF. Where status ON means light in “on” and OFF indicate the absence of light signal. Quantitative values for each qualitative status can be defined as;

$$F_I = \begin{cases} 1 & \text{Indicator is ON} \\ 0 & \text{Indicator is OFF} \end{cases}$$

In addition to the Indicator, two quantitative features of surrounding vehicles are considered - *distance* and *orientation*. Both the features are kept in an array, Where a distance array F_D consists of the distance of surrounding vehicle from the subject vehicle and another F_O holds orientation information.

$$F_D = [F_{D1}, F_{D2}, \dots, F_{Dn}]$$

$$F_O = [F_{O1}, F_{O2}, \dots, F_{On}]$$

2.2 Intention Recognition

Navigation intent can be determined by the function, structure and behavioural aspect of the environment object [18]. The behaviour of taking a turning or going straight could be one of the functional property of surrounding vehicles, which is captured in a fuzzy set of having membership for each such surrounding vehicle. A Fuzzy based Intention prediction is used to capture the intention of all the neighbour vehicles via two membership functions τ_t and ξ_t , where the functions are related to each other. ξ_t determines the membership value for moving straight and τ_t represent membership value for taking turn. ‘ α ’ is rate of change of membership functions with respect to Acceleration and ‘ β ’ is rate of change of membership functions with respect to Indicator signal. ‘ a ’ is acceleration of neighbour vehicle.

$$\xi_t = \begin{cases} 1 & t = 0 \\ 1 - \tau_t & F_I = 0 \\ \beta\xi_{t-1} & F_I \neq 0 \end{cases}$$

Membership value of going straight is maximum and membership value of turning is minimum initially as there is no need to take turn without any obstacle in the path. Membership value of going straight is changing with respect to turn by subtracting the τ_t from the maximum value. Here α is greater than β because in this scenario impact of indicator is much more than the impact of variation in speed.

$$\tau_t = \begin{cases} 0 & t = 0 \\ 1 - \xi_t & F_I \neq 0 \\ \frac{\tau_{t-1}}{\alpha} & F_I = 0 \wedge a > 0 \\ \alpha\tau_t & F_I = 0 \wedge a < 0 \end{cases}$$

If a surrounding vehicle intends to come into the path, i.e. indicator signal F_I is ON, indicating the possibility of crossing the way of automated vehicle then membership function of going straight is decreased by the factor β and membership value of turning is increased by subtracting the ξ_t from maximum value.

Membership values also varies with acceleration of neighbour vehicle as if neighbour vehicle is retarding down their is a possibility of taking turn and it may come into the path of automated vehicle so membership value of turn for automated vehicle should be increased by the factor inverse of α . If neighbour vehicle is speeding up (i.e. accelerating) then membership value of taking turn is decreased by the factor α .

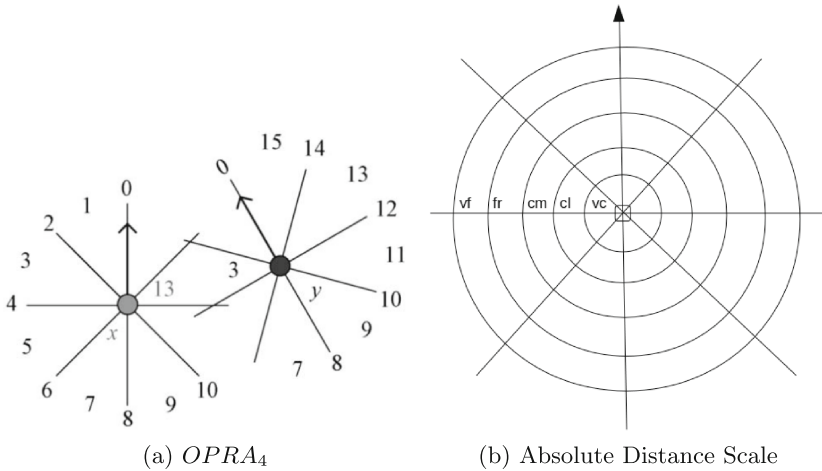


Fig. 2. Representation of orientation and direction scheme. **(a)** A basic relation in $OPRA_4$. **(b)** A combined illustration of orientation and direction.

2.3 Orientation and Distance

Apart from the prediction of intention of surrounding vehicle based on a qualitative feature, orientation and distance would also play a significant role. For orientation information the Orientation Point Algebra (*OPRA*) [14] is used to describe the relative direction information, where $OPRA_m$ signifies the uses of m number of lines going through the object point and can be visualised in Fig. 2(a), in which orientation point x lies on the third part of the space divided by lines going through oriented point y , whereas y lies on 13th part of space divided by the lines going through x under the assumption of $m = 4$.

Distance in spatial domain can represent by either absolute scale or some relative measurement. We consider distance on an absolute scale where notions such as *very close*, *close*, *commensurate*, *far*, and *very far* could be used. In general, the distance relation has meaning only when combined with direction relation. Therefore distances are used together with *OPRA*.

3 Implementation and Results

The objective to design a fuzzy model was to analyze the effect of intention based method in navigation. Where we are interested only on the path obtained by the autonomous agent in the different scenario. Therefore, instead of exhaustive implementation and considering many features and real scan data, a sample set of data is used in Matlab to fulfill the objective. Demonstrative results of the path taken by the integrated autonomous vehicle in different circumstances are shown. A comparison of this integrated approach with potential field navigation method is done for the analysis. Results represented here can broadly divide into two categories *Navigation Path* in different surrounding scenarios and *Comparison and Analysis* with other existing methods.

3.1 Navigation Path

A different surrounding environment scenario requires a different navigation path strategy. Autonomous navigation vehicle should follow the navigation according to the present surrounding scenario. It should avoid the collision and achieve the goal by keeping the motion smooth. Few scenarios and respective navigation paths are demonstrated in Fig. 3, which presents the navigation path in presence of one vehicle in the surrounding and shows the path followed by the autonomous vehicle with different values of the effective range. Where the effective range is that distance from which vehicle starts observing the surrounding objects. Figure 3(a) shows the path of the vehicle when the effective range is large where Vehicle observed the presence of a surrounding vehicle and starts taking a curve turning to avoid the sharp turn. This proactive observation gives machine more time to take the turn and make the motion smoother. Similarly, in Fig. 3(b), effective range is medium and the vehicle starts taking turning after covering some distance from the initial point. Figure 3(c) shows a small value of range and observation of surrounding vehicle starts when they are very close. In this scenario vehicle gets a very small time to take action and path becomes curved.

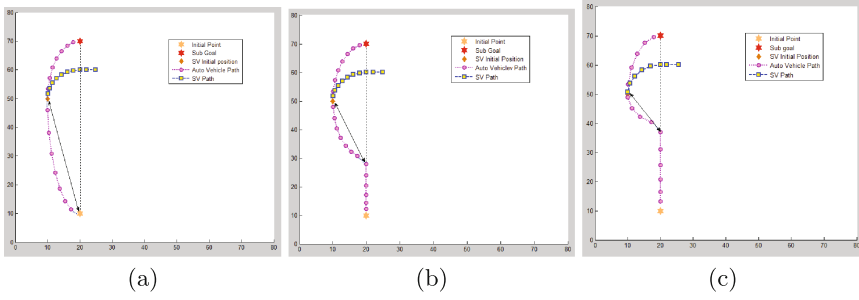


Fig. 3. Navigation Path for different effective range of Autonomous Vehicle. (a) Large effective range. (b) Medium effective range. (c) Small effective range.

3.2 Avoiding Obstacle

Figure 4(a) shows the navigation path followed by the subject vehicle when one obstacle is present, where it observe the surrounding vehicle at initial point

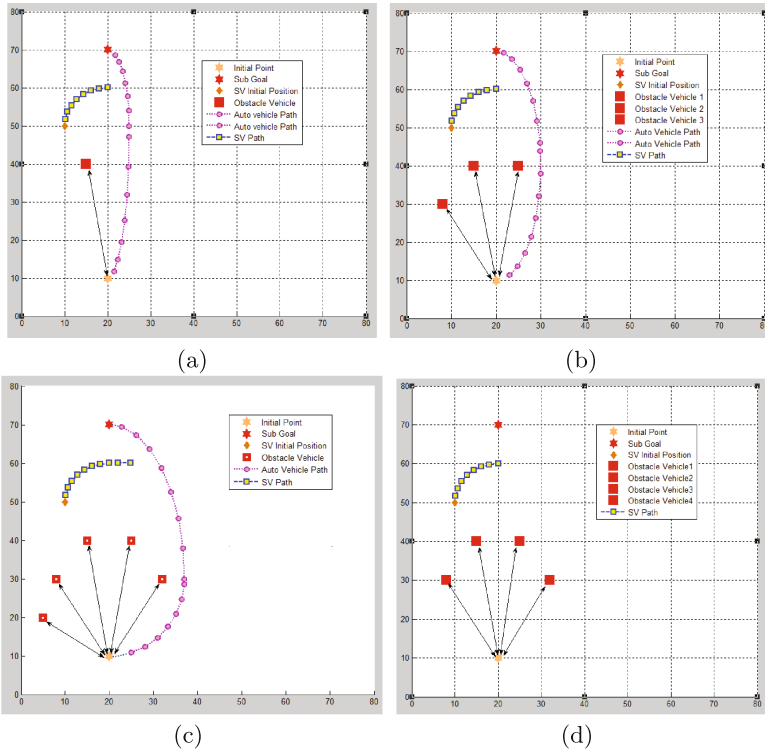


Fig. 4. (a) shows the path followed by Autonomous Vehicle with one obstacle. (b) shows the path followed by Autonomous Vehicle with three obstacles. (c) shows the path followed by Autonomous Vehicle with five obstacle. (d) shows the Vehicle can not move forward due to obstacles.

and starts turning to avoid it but at the same time it observe the indication of turning intention of an another vehicle, which command to calculate a new path. Therefore it select the control from the decision making module to predict the next optimal path and pass it to the action module. Multiple obstacles can also be their as shown in Fig.4(b) and (c), where it every time when an obstacle is found in the path it calculate the new optimal path with the help of decision tree. Figure 4(d) represents a different situation where all possible directions are obstructed by the obstacles and vehicle has no way but to stop. In this situation decision maker will return all possible directions one by one and if it will not find any clear path then it signals the stop command to the vehicle.

4 Analysis

Although exact comparison could be made only in the dynamic environment, a comparison study is shown here to give an idea to differentiate between the results of reactive and proactive approach. Here potential field navigation

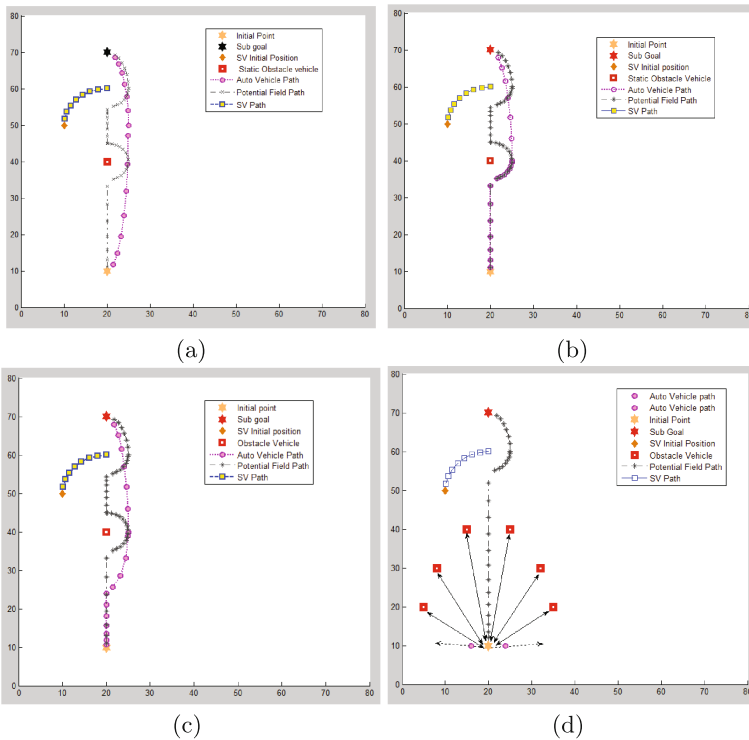


Fig. 5. Path obtained (a) when effective range is large (b) when effective range is small (c) when effective range is medium (d) shows the Vehicle can not move forward due to obstacles.

method is compared with the presented method. Figure 5(a) represents the comparative path of both the methods. Line with crosses shows the path followed by potential field navigation where the other line with circles shows the path of new proactive approach. Effective distance taken here is large and it is clear from the graph that new approach gives a much smooth path with less number of curves. This approach becomes closer to the reactive approaches as the size of effective distance is decreased as shown in Fig. 5(b). As the size of effective range is decreased the proactive power of vehicle also decreased. Small effective range leads to the late prediction of intents of surrounding vehicles hence reduces the pro-activeness. Vehicle follows the same path as followed by potential field method because in this case intents of surrounding vehicle (SV) can be calculated at approximately same time when it comes into the path of vehicle. So both the techniques take turn at same time. Apart from this Fig. 5(a) represents the comparison with a medium value of effective range. In this case path produced by novel approach has lesser number of curves and it avoids the obstacles more smoothly as shown in graph.

5 Conclusion

In this paper integration of intent recognition with decision making for navigation assistance of a mobile robot have been presented. Implementation of the approach strengthens the claim to consider intention based decision making for mobile robot navigation. Such a framework can predict the future course of action much before the reactive systems. Conclusively, it can be observed that the approach proposed in this paper has many advantages over the existing reactive techniques. Nevertheless, there are certain scenarios where reactive methods may perform better. Implementation within a robotic platform like ROS (Robotic Operating System) may provide a better way to evaluate the claim. This is part of on-going research.

References

1. Andrews, J.R., Hogan, N.: Impedance control as a framework for implementing obstacle avoidance in a manipulator. Master's thesis, M. I. T., Dept. of Mechanical Engineering (1983)
2. Bader, S., Kirste, T.: A Tutorial Introduction to Automated Activity and Intention Recognition. Lecture Notes for the Interdisciplinary Colleg, IK (2011)
3. Barshan, B., Durrant-Whyte, H.F.: Inertial navigation systems for mobile robots. *IEEE Trans. Robot. Autom.* **11**(3), 328–342 (1995)
4. Borenstein, J., Koren, Y.: The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Trans. R A* **7**(3), 278–288 (1991)
5. Chen, J., Cohn, A.G., Liu, D., Wang, S., Ouyang, J., Yu, Q.: A survey of qualitative spatial representations. *Knowl. Eng. Rev.* **30**(01), 106–136 (2015)
6. Clementini, E., Di Felice, P., Hernández, D.: Qualitative representation of positional information. *Artif. Intell.* **95**(2), 317–356 (1997)

7. Elfes, A.: Using occupancy grids for mobile robot perception and navigation. *Computer* **22**(6), 46–57 (1989)
8. Garcia-Cerezo, A., Ollero, A., Aracil, J.: Stability of fuzzy control systems by using nonlinear system theory. *Annu. Rev. Autom. Prog.* **17**, 121–126 (1992)
9. Goodrich, M.A., Schultz, A.C.: Human-robot interaction: a survey. *Found. Trends Hum. Comput. Interact.* **1**(3), 203–275 (2007)
10. Hofmann, A.G., Williams, B.C.: Intent recognition for human-robot interaction. In: *Interaction Challenges for Intelligent Assistants*, pp. 60–61 (2007)
11. Ishikawa, S.: A method of indoor mobile robot navigation by using fuzzy control. In: *IEEE/RSJ International Workshop on Intelligence for Mechanical Systems Intelligent Robots and Systems 1991*, pp. 1013–1018 (1991)
12. Kelley, R., Tavakkoli, A., King, C., Nicolescu, M., Nicolescu, M.: *Understanding Activities and Intentions for Human-Robot Interaction*. INTECH Open Access Publisher, Rijeka (2010)
13. Li, T.-H.S., Chang, S.-J., Chen, Y.-X.: Implementation of human-like driving skills by autonomous fuzzy behavior control on an FPGA-based car-like mobile robot. *IEEE Trans. Ind. Electron.* **50**(5), 867–880 (2003)
14. Moratz, R., Dylla, F., Frommberger, L.: A relative orientation algebra with adjustable granularity. In: *Proceedings of the Workshop on Agents in Real-Time and Dynamic Environments (IJCAI 2005)*, pp. 61–70 (2005)
15. Ollero, A., García-Cerezo, A., Martínez, J.L.: Fuzzy supervisory path tracking of mobile reports. *Control Eng. Pract.* **2**(2), 313–319 (1994)
16. Saikia, A., Khan, M.A., Pusph, S., Tauhidi, S.I., Bhattacharyya, R., Hazarika, S.M., Gan, J.Q.: cbdi-based collaborative control for a robotic wheelchair. *Procedia Comput. Sci.* **84**, 127–131 (2016)
17. Shalin, V.L., Perschbacher, D.L., Jamar, P.G.: Intent recognition: an emerging technology. In: *Proceedings of the International Conference on Human-Machine Interaction and Artificial Intelligence in Aeronautics and Space, Toulouse-Blagnac*, pp. 125–137 (1988)
18. Thompson, S., Horiuchi, T., Kagami, S.: A probabilistic model of human motion and navigation intent for mobile robot path planning. In: *Autonomous Robots and Agents, ICARA 2009*, pp. 663–668. IEEE (2009)
19. Thrun, S., Fox, D., Burgard, W., Dellaert, F.: Robust Monte Carlo localization for mobile robots. *Artif. Intell.* **128**(1), 99–141 (2001)

Clinical Intelligence: A Data Mining Study on Corneal Transplantation

Brian Carneiro, Rui Peixoto, Filipe Portela ^(✉), and Manuel Filipe Santos

Algoritmi Research Centre, Universidade do Minho, Campus Azurém,
4800-058 Guimarães, Portugal
cfp@dsi.uminho.pt

Abstract. The purpose of this study is to analyse the Oporto Hospital Center (CHP) corneal transplantation process using Data Mining (DM) techniques, following the Cross Industry Standard Process for Data Mining (CRISP- DM) methodology. The DM goals focused on the definition and evaluation of DM models capable of predicting the priority of a request for a surgical procedure and its waiting time. Thus, 320 models were generated using the Pervasive Data Mining Engine (PDME) tool. The model results showed that although there is no model capable of effectively predicting all priority target classes, a “normal” class can be used to accurately perform this type of prediction, due to good sensitivity results. In some models, the sensitivity achieved results of 94% or even 99% along with an accuracy slightly over 80% for a specific target class.

Keywords: Corneal transplantation · Clinical intelligence · Data mining

1 Introduction

The cornea is a transparent, avascular and elastic tissue placed in the anterior part of the eyeball, being considered the most powerful lens in the visual system [1]. Cornea maintains the intraocular pressure, supporting the internal structures of the eye and resisting trauma [2]. Diseases affecting the cornea are a major cause of blindness, second only to cataract in overall importance [3]. Corneal transplantation remains the primary sight restoring method for corneal blindness as well as is considered the most frequently performed type of transplant worldwide [4].

According to Borges et al. [5], the CHP began corneal transplants in 1958, at a time when few centers in Europe were performing. The CHP is considered a reference in the field of transplantation, being the first cornea transplant program in Portugal to be certified. Due to the importance of this type of transplantation to the CHP, a better comprehension of the corneal transplantation process was required. As traditional database systems are not sufficient for proper analysis of health data, a Clinical Intelligence (CI) system was designed. This system allows the integration and transformation of ambiguous, incomplete and inconclusive raw data, into knowledge. This knowledge will provide timely information and insights, thus contributing to an effective information platform for decision makers. To foster understanding of the corneal transplantation process, this study aims to analyze the corneal transplantation process using DM

techniques. By using the CI platform to collect and integrate the data, the DM will provide a better understanding of how the process variables relate and thus unlocking new information. The DM process will follow the CRISP-DM methodology and will have as DM objectives, the definition and evaluation of DM models capable of predicting the priority of a surgical procedure application as well as the patient's waiting time. The DM tool adopted is the PDME, which allows the automatic execution of DM processes, the construction of parallel models, the registration and comparison of all instances of the process [6].

This paper is divided into six sections: Introduction; Background; Methods and Tools; Case Study; Discussion; Conclusion and Future Work. The second section presents a description of the cornea as well as a brief explanation of the CI system developed and the related work. In Sect. 3, the methods and tools utilized for this project are presented and described. In Sect. 4 the case study is presented, following the CRISP-DM process flow, including data comprehension, preparation, modeling, and evaluation. In Sect. 5 the results are evaluated in the project context. Finally, in Sect. 6, a summary of this paper is given, describing the identification of the main discoveries and a reflection of the whole process. In addition, a short description of the future work is presented.

2 Background

2.1 Cornea

According to Sousa [1], the cornea is the first and most powerful lens of the eye's optics, accounting for about 70% of the total refractive power of the eye's dioptric system. In addition to the optic property, the cornea performs three mechanical functions: maintaining intraocular pressure together with the sclera, supporting the internal structures of the eye and resisting trauma. Macroscopically, the cornea is a transparent, avascular, fibrous membrane that lies in the anterior opening of the sclera presenting a greater curvature than the rest of the eyeball. Microscopically, the cornea is divided into five layers: epithelium, which contributes to the maintenance of the optic surface of the cornea. The layer membrane, whose layer does not regain after injuries. Stroma, whose anatomical and biochemical properties ensure the transparency, strength, and stability of the cornea. Descemet's membrane and the endothelium [3].

2.2 Clinical Intelligence System

A CI system was developed to analyze the process of CHP's corneal transplantation, based on the set of business indicators and the data provided by the organization. The dataset contained a sample size of 428 pairs of eyes, through the period of 2013–2016. The system, which development followed Kimball's Lifecycle Methodology, is divided into three main environments: Data sources, Development and Visualization. In Fig. 1, the architecture is presented, identifying the system flow process as well as the tools adopted.

- **Data Sources:** The data provided by CHP in an excel format.

- **Development:** Initially, the extraction, transformation, loading (ETL) process occurs with the extraction of the data into this environment. Then the data transformation task is executed through procedures, such as data cleaning and aggregation or the removal of duplicated data. Afterwards, the transformed data are loaded to the data warehouse dimension models, on a star dimensional format. Next, the Online Analytical Processing (OLAP) tabular cube was developed.
- **Visualization:** A set of reports are provided in a Power BI interface.

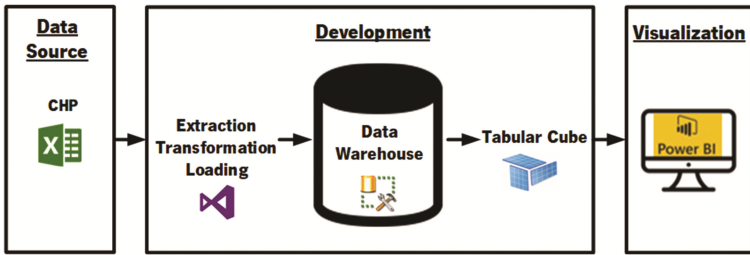


Fig. 1. CI architecture

The CI system provided a better understanding of the corneal transplant process evolution regarding the number of surgeries applications and patients waiting time, along with the identification of trends of procedures, diagnostics, anesthetics and main patient's characteristic. The implementation of this technology assured the quality and the integration of the extracted data sources, along with the efficient data manipulation, which allowed new insights to physicians and management, to support better and informed decision making.

2.3 Related Work

In 2016, a study on the evolution of corneal transplants at the CHP was carried out [5]. This study presented a retrospective analysis of all corneal transplants performed at the CHP from January 2005 to June 2015. Although the work analysed the same corneal transplantation process as the present study, it was not carried out using data mining techniques as well as CI solutions such as tabular cubes. This combination of tools and techniques allowed greater manipulation, management and intuitive analysis of the data as well as relevant process predictions, hence providing to the physicians and CHP management, a complete analysis and a platform for informed decision making.

3 Methods and Tools

For the project as whole, the Design Science Research (DSR) was followed as it presents a clear definition of the outputs and flow of the process as well the flexibility between phases. [7]. Nevertheless, the present study is inserted in the "design and development phase", as it provides an additional analysis and solutions to the CI system previously

developed. In order to perform the DM analysis, the CRISP-DM was chosen as the DM methodology [8]. The adoption of this methodology is because it is a globally accepted framework, with a clear description of the tasks in each phase and the flexibility between them. Below is presented the CRISP-DM life cycle in the context of this study.

- **Business understanding:** Comprehension of the CHP objectives and requirements, to define the data mining problem and the work plan.
- **Data understanding:** An exploration of the two dataset files as well as performing data quality assessment.
- **Data preparation:** Design of the final dataset using the CI system ETL process.
- **Modeling:** Application of classification techniques to the ten scenarios designed, as well as the configuration of parameters in the PDME tool
- **Evaluation:** Review of the designed model's results, in order to verify if the business objectives were achieved.
- **Deployment:** In this study, this phase was not performed, since the purpose of this study did not imply the implementation of the DM process in CHP.

In order to execute the designed models, the PDME tool, which is based on R programming language, was used. According to Peixoto et al. [6], the PDME was developed to facilitate the use of DM engines, which require optimizations by data mining experts in order to provide optimal results. PDME adds pervasive characteristics, such as invisibility and ubiquity, improving the user experience as well providing autonomous and intelligent DM processes. The adoption of this tool was because it presented benefits such as the automatic realization of DM processes, the friendly user interface, the construction of DM models in parallel, the registration of all instances of the process and the possibility of comparison between them.

4 Case Study

4.1 Business Understanding

Given the importance of the corneal transplantation for CHP, the need arose for a better understanding of its process. Thus, the goal of this study is to analyze the transplantation process using DM classification techniques. This study will assess how certain targets can be accurately predicted, contributing to better management and physician's efficiency. In this sense, the DM objectives are the definition and evaluation of DM models capable of predicting the surgical procedure priority and its waiting time.

4.2 Data Understanding

The data, provided by the CHP, are organized in two Excel files. The first file, Cornea (C) contains 472 records and 28 attributes, regarding the process of managing the corneal transplant waiting list. The second file, Surgery (S) has 1319 records and 31 attributes describing the surgical intervention elements. Table 1 maps the variables used as input in the ETL process with the data sources along with the variables description.

Table 1. Variables description

Attribute	C	S	Description
NUM_PROCESSO	X	X	Patient Hospital Process Identification Code (ID)
NUM_Sequential	X	X	Patient entry on a specific area ID
COD_RISCO_ANEST		X	Patient Fitness before surgery
SEXO	X		Patient Gender
DTA_NASCIMENTO	X		Patient Birth Date
EPISODIO	X	X	Patient hospital entry in any area
CID_DIAGBASE	X	X	ID of the Patient Diagnostic
CID_PROCIR	X		Procedure ID
LOC_PROC	X		Eye to be intervened
PRIORIDADE	X		Surgery priority
COD_ANESTESIA	X		Anesthesia ID
CON_ANEST	X		Pre-Anesthesia Consultation
DTA_PED	X		Surgery Application Date
DTA_SAIDA	X		Exit date

Afterwards, a data exploration was performed to obtain a better understanding. Therefore, regarding the episodes collected in the first file, 51.27% are associated with female patients while male occurrences are slightly smaller and correspond to 48.73%, which demonstrate the balance between these two classes. The maximum age identified is 98 years, and the minimum age corresponds to a baby with a certain age of 1 month. Figure 2 presents the percentage distribution of each type of surgery. Penetrating Keratopathy and Posterior Lamellar procedures are the most frequent surgeries performed, with 42% and 41% respectively. By a significant margin, the anterior Lamellar (17%) is the third most performed procedure in CHP.

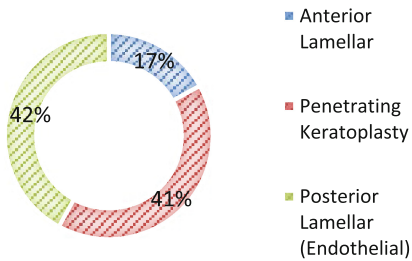


Fig. 2. Type of procedure distribution

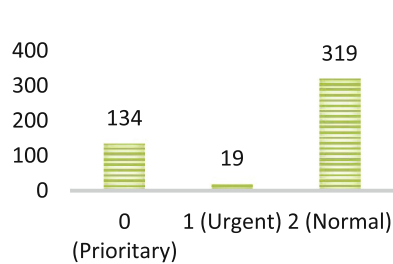


Fig. 3. Surgery priority distribution

As for the priority procedures (Fig. 3) which represent a target for the DM model, there is a huge discrepancy between the three classes. The most common occurrences are the normal applications, as expected, with 68%, followed by the priority requests (28%) and then the urgent cases, with only 4%. In order to complete this phase, a data quality assessment was conducted. This assessment contributed to the definition of correction strategies that may have an impact on DM models. In general, in both datasets,

it was possible to identify the following inconsistencies: attributes with null values; numerical attributes with text; misspelling or identifier codes with different associated descriptions.

4.3 Data Preparation

Initially, the two datasets were integrated through the sequence number of each existing record in both files, thus generating a new dataset with 425 records. Subsequently, the pertinent attributes were selected according to the objectives outlined by the CHP. This new dataset was then subjected to an ETL process, using the previously conceived CI platform integration system (Fig. 4).

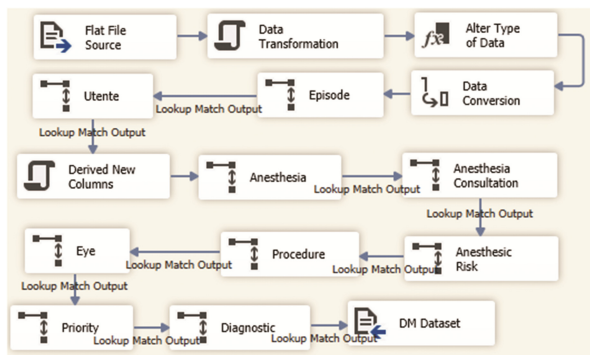


Fig. 4. ETL process

As mentioned above, this ETL process starts with the extraction of the dataset for the application, and then the data transformation phase emerges. In general, the data transformation consisted of the following actions: treatment and grouping of the different attribute values; standardization of the surgery application and exit date (day-month-year); deriving three new attributes. Afterwards, lookups are performed in order to, first, verify if a certain attribute exists on the platform database, second to identify which attribute will appear in the final dataset. In the end, a new dataset is generated in a CSV format. It should be noted that different iterations of this process were created in order to adjust the data to the different scenarios conceived.

4.4 Modeling

The ETL process generated different test datasets according to the objectives defined for each model. The attributes contained in these datasets are:

- **VA:** Gender: describes the patient's gender
- **VB:** Diagnostic: identifies the cornea disease
- **VC:** Priority: classifies the Procedure Application
- **VD:** Surgery: describes the type of Procedure

- **VE:** Eye: identifies which eye will be intervened
- **VF:** Risk Anesthesia: assesses the patient's fitness
- **VG:** Pre-Anesthesia Consultation: identifies If the patient required pre-Anesthesia consultation
- **VH:** Anesthesia: classifies the type of the anesthesia
- **VI:** Waiting Time: defines the time between the procedure application and the patient exit.

With the testing datasets designed, the objective was to predict the priority of the patient's request for a procedure and the respective waiting time. Thus, different scenarios and groups of variables were submitted to classification algorithms with the PDME tool. Thus, the following scenarios were created:

- S1, S2, S3 = {VA, VB, VC, VD, VE, VF}
- S4, S5 = {VA, VB, VC, VD, VE, VF, VG, VH, VI}

In Scenario 2 (S2), the sampling technique was applied to the data in order to balance the target values, in this case the duplication of the data, since the imbalance or dominance of one value under another can cause the minority of the data to present a greater number of errors. In scenario 3, a new grouping of values in the "diagnostic" variable was performed in order to balance the data as well. Scenario 4 and 5 have a different target than the previous scenarios, and while the first, has a target with three classes, the other only has two. These scenarios were submitted to the following thirty-two classification algorithms:

- e1071_naiveBayes; caret_hdda; caret_lda; caret_rocc; LiblineaR; C50; caret_rf; caret_fda; caret_rpart; caret_JRip; J48; e1071_rpartC; caret_ctree2; caret_nnet; caret_C50; caret_RSimca; caret_deepboost; caret_gcvEarth; caret_earth; LMT; caret_gbm; e1071_randomForest; caret_RRFglobal; caret_bayesglm; caret_PART; randomUniformForest; caret_RRF; parallelSVM; caret_rotationForest; kernlab; e1071_svm; caret_adaboost.

Scenarios 1, 2 and 3 can also be represented as DMM = {3 scenarios; 32 Techniques, 2 Targets} generating 192 models, while scenarios 4 and 5 can be described as DMM = {2 scenarios; 32 Techniques, 2 Targets}, providing 128 models. Overall, 320 models were created through the PDME tool. The configuration of PDME is particularly relevant when the user, first, has to define the scores and measures (ACC, Kappa, F1, Precision, True Positive Rate and True Negative Rate) for the classification evaluation, second, when the data partition percentages definition is needed (K-fold and Holdout).

4.5 Evaluation

Table 2 presents the models and scenarios (S) evaluation using the following metrics: Sensitivity (ST), Accuracy (A) and Specificity (SP) according to the CHP.

Table 2. DM model results

Target	Scenario	Class	A	SP	ST
Procedure priority	1	Normal	83%	67%	95%
		Priority_Urgent	95%	99%	67%
	2	Normal	75%	75%	84%
		Priority_Urgent	76%	85%	75%
	3	Normal	80%	82%	82%
		Priority_Urgent	78%	79%	82%
Waiting time	4	<10	84%	97%	86%
		>=10-<120	78%	96%	85%
		>=120	80%	73%	99%
	5	<120	90%	94%	77%
		>=120	82%	77%	93%

In scenario 1, the “Normal” class has an accuracy of 83%, a specificity of 67% and a sensitivity of 95%. Therefore, the model can predict this class in a relatively accurate way and to identify the priority of a procedure request, but it has difficulties in identifying false positives. Regarding the “Priority_Urgent” class, the model is very accurate (95%), but unlike the previous class, the model has difficulties in predicting the correct procedure priority. In Scenario 2, with the sampling of the data, the accuracy has a low accuracy and specificity with 75%, however it can identify consistently true positives (85%) for the first class. The “Priority_Urgent” class has a similar accuracy but has a greater capacity to identify false positives than the true positives (85 and 75%). Scenario 3 presents a model with consistent and relatively accurate results for both target classes (80% and 78%). There is little discrepancy between the target classes at a specificity and sensitivity level, with values around 82%.

In scenario 4, which addresses the “Waiting Time” target the accuracy of the model is balanced between the different classes of the target. In the “<10” class, the model is relatively accurate (84%) and able to identify true positives (86%) as well as false positives (97%). For the class “>10 && <120”, the model shows good results in terms of specificity and sensitivity, with 85% and 96% respectively, however the model is not very precise (78%). The Class “>=120” presents low acuity and specificity (80% and 73%), however it is necessary to identify the real priority of the waiting time (99%). In scenario 5 the model presents particularly accurate results. For the class “<120”, the model is very precise and able to clearly identify false positives, with 90 and 94% respectively, however its sensitivity is already much lower (77%). The class “>=120” presents low results, in terms of acuity and especially specificity, with 82% and 77% respectively. With 94% sensitivity, the model can identify, for this class, the actual waiting time precisely.

5 Discussion

As stated before, the models are evaluated using the following metrics (by order of importance): Sensitivity, Accuracy, and Specificity. For the CHP, a model is considered

acceptable if the following conditions are observed: Sensitivity ≥ 0.85 ; Accuracy ≥ 0.75 . In this evaluation, the sensitivity measure assumes the value 1 and the specificity the value 0. Overall, these results demonstrated that for the surgery priority target there is no model acceptable to the CHP standards considering target classes as a whole. In fact, few target classes are presenting simultaneously acceptable CHP accuracy and sensitive results. Nevertheless, the models for this target can still be used and provide great assistance to physicians and management, if the focus is on a specific class, as the models present good results for the sensitivity measure. As an example, in scenario 1 for target “procedure priority”, the model can clearly predict the class “normal”, with a 95% sensitivity and an 83% accuracy. This class represents the most common applications in the CHP. Regarding the waiting time target, there is in fact a model that achieves the CHP results standard, allowing to perform an accurate prediction. As for negatives notes, the models with low values of specificity can generate false positive situations. Nevertheless, this aspect is irrelevant considering the fact it detects the true positive cases accurately. When the prediction is wrong, the patient health can be jeopardized, so an accurate prediction is crucial.

In summary, these models proved that is indeed possible to predict the certain targets classes accurately as it presents good results for sensitivity. These models will also allow CHP to manage the patient condition accordingly as well as the CHP resources. Furthermore, the physicians will mainly consider the sensitivity results to predict possible transplantations priorities. In note, the model and data validation were performed, and later this process will be assessed by the physicians.

6 Conclusions and Future Work

In virtue of this study, it was possible to analyze the process of the corneal transplants using DM techniques. The purpose was to predict the surgery priority and the respective waiting time. Using the CI system, the ETL process generated different test datasets according to the objectives defined for each model. Based on the 320 generated models, it was possible to verify that an accurate prediction can be performed, if the focus is on the waiting time target, as no model had acceptable conditions for every priority target class, according to the CHP conditions (Sensitivity ≥ 0.85 and Accuracy ≥ 0.75). Nevertheless, these models are useful to predict certain classes, as the sensitivity results, can achieve 94 or even 99 percent and hold an acceptable accuracy. In other words, although the results are not good as a whole, they are indeed good predicting the certain target values due to its sensitivity. Therefore, the CHP can properly predict the surgery application priority and the patients waiting time, hence allowing to manage the patient's condition accordingly, coordinate the hospital resources as well as the corneal transplantation waiting list process more efficiently.

Regarding future work, it is expected that the CHP provides more data sources, hence allowing to increase and add more historical context. Proven the benefits of the models, the Data Mining process will be incorporated into the CI system. In the end, the system will be implemented in the CHP.

Acknowledgement. This work has been supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT – Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2013. This work is also supported by the Deus ex Machina (DEM): Symbiotic technology for societal efficiency gains - NORTE-01-0145-FEDER-000026.

References

1. de Sousa, C.I.P.: Caracterização dos parâmetros da película lacrimal e da topographic corneal na população adulta portuguesa: um estudo piloto. University of Minho (2014)
2. Otel, I.: Evaluation of Corneal Nerve Morphology for Detection and Follow-up of Diabetic Peripheral Neuropathy. University of Coimbra (2012)
3. Frigo, A.C., et al.: Corneal transplantation activity over 7 years: changing trends for indications, patient demographics and surgical techniques from the Corneal Transplant Epidemiological Study (CORTESE). *Transpl. Proc.* **47**(2), 528–535 (2015)
4. Gain, P., et al.: Global survey of corneal transplantation and eye banking. *JAMA Ophthalmol.* **134**(2), 167–173 (2016)
5. Borges, T., Lages, V., Coelho, J., Gomes, M., Oliveira, M.: Evolução dos Transplantes de Córnea no Centro Hospitalar do Porto: Da Queratoplastia Penetrante aos Transplantes Lamelares. *Rev. Soc. Port. Oftalmol.* **40**(4) (2017)
6. Peixoto, R., Portela, F., Santos, M.F.: Towards a pervasive data mining engine—architecture overview. In: Rocha, Á., Correia, A., Adeli, H., Reis, L., Mendonça Teixeira, M. (eds.) *New Advances in Information Systems and Technologies*. AISC, vol. 445, pp. 557–566. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31307-8_58
7. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. *J. Manag. Inf. Syst.* **24**(3), 45–77 (2007)
8. Chapman, P., et al.: CRISP-DM 1.0 Step-by-Step Data Mining Guide (2000). <https://www.the-modeling-agency.com/crisp-dm.pdf>

High-Quality Medical Image Compression Using Discrete Orthogonal Cosine Stockwell Transform and Optimal Integer Bit Allocated Quantization

Vikrant Singh Thakur¹(✉), Kavita Thakur², and Shubhrata Gupta¹

¹ Department of Electrical Engineering, National Institute of Technology, Raipur 492010, India

vikrant.st@gmail.com

² S.O.S. in Electronics and Photonics, Pt. Ravishankar Shukla University, Raipur 492010, India

Abstract. Communication of the medical image and videos has now raised as a vital concern for the telediagnosis of critical diseases. Currently, JPEG and JPEG2k codecs are the default compression tool to facilitate their communication over band-limited channels. However, most often, the performance of these existing codecs is found poor particularly at the higher compression levels. Hence, this paper presents a new medical image compression codec to achieve high-quality compression of the medical images, especially at the higher compression levels. The proposed codec utilizes Discrete Orthogonal Cosine Stockwell Transform (DOCST) for the higher pixel decorrelation and the optimal integer bit allocation based quantization strategy for the efficient quantization of the DOCST coefficients. Further, to justify and validate the performance of the proposed codec an extensive performance analysis has been presented for six medical images of two different modalities. It is reported that the proposed codec outperforms the existing JPEG and JPEG2k codecs with significant quality gain for all the compression levels.

Keywords: Medical image compression · Image transform codecs
DOCST transform · Optimal integer bit allocation · JPEG
JPEG2k · Image quality

1 Introduction

In the present scenario, almost all the hospitals are highly equipped with the modern setups to avail telediagnosis and telemedicine facilities from the various prominent specialists. Meanwhile, due to the rapid enhancements in the medical imaging systems, they are now capable of acquiring high dimension slices with sound spatial resolutions to provide minute information related to the diseases [1]. Consequently, day-by-day, the extent of medical data generated by the hospitals is exponentially increasing. Hence, for the telemedicine and telediagnosis process, the transmission of such high dimension images through bandwidth-limited channels and their storage requirements

have become an arduous problem. To properly handle these issues some efficient medical image compression and their improved transmission techniques are of utmost importance. One of the most important medical image compression methods is image transform coding technique [2]. The popular existing codecs for the medical image compression, like Joint Photographic Experts Group (JPEG) codec and JPEG2k codec are all developed on the concept of transform coding technique [3–7]. These exciting standard codecs are currently serving to deliver a good quality medical image compression against the other available codecs. However, the increased dimension of medical images, now demanding very high compression or equivalently compression at low bits-per-pixel (bpp) levels for their transmission and storage. At the lower bpp levels, the available JPEG and JPEG2k codecs mostly offer poor quality image compression. Therefore, the present context of medical image compression aspects some new advanced codecs which can able to deliver good quality compression over low and very low bpp levels. To address this difficulty, in this paper, the authors present an advanced medical image compression codec, which utilizes a recent image transform Discrete Orthogonal Cosine Stockwell Transform (DOCST) for higher energy compaction [8, 9]; and an optimal integer bit allocation strategy to achieve efficient quantization environment compared to existing uniform scalar quantizers.

The rest of the paper is organized as follows. Section 2 presents the designing aspects of the proposed medical image codec. Section 3 shows the experimental results of the present work. Finally, Sect. 4 presents the conclusive remarks of this work.

2 Proposed Medical Image Codec

The ultimate design goal behind the development of an image compression codec is to deliver the best visual and objective quality over highest possible compression levels. The term compression level generally referred in terms of compression rate measured in bits-per-pixel (bpp); which defines the average number of bits used to encode the pixels of the image [2].

As we have seen in the previous section, currently, the image transform coding technique based codecs JPEG and JPEG2k are serving to deliver best possible visual quality among the existing compression codecs. However, it is practically a tough task to satisfy both the quality and the bitrate tradeoffs. Consequently, the existing codecs often fail to maintain the quality and bitrate tradeoffs at the higher compression or equivalently lower bpp levels, resulting in poor quality image reconstruction. Therefore, to design the proposed medical image coder, the authors first investigate the general structure of the image transform coding technique. In general, a transform codec performs three main functions in the encoding process [2]:

- (1) Image transformation to compact the signal energy.
- (2) Quantization of the transform coefficients.
- (3) Lossless entropy coding to generate a compressed bitstream.

Among the above three functions, the first two are much important and hence the performance of the transform codecs are highly depending on these two functional blocks. Therefore, for the first part of the proposed codec design, a recently developed

Discrete Orthogonal Cosine Stockwell (DOCST) transform has been utilized to achieve higher energy compaction than the popular Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT). Whereas, in the second part an optimal bit allocation strategy based quantization technique is employed to efficiently quantize the DOCST transform coefficients. Since, among the above three functions, the entropy coding part is a lossless strategy to map the quantized transform coefficients into bit-stream. Hence this part has been removed from the proposed coder and the final compression rate is simply determined through the first order entropy.

The first order entropy is a simple measure of an average number of bits per pixel required to encode the complete image. Suppose a discrete random variable r_k defined in interval $[0, L - 1]$, represents the intensities of an $M \times N$ image, then with the occurrence probability of $p_r(r_k)$, the first order entropy for the image is defined as,

$$H = - \sum_{k=0}^{L-1} p_r(r_k) \log_2 p_r(r_k) \quad (1)$$

Further, a brief description of the design process for the first two blocks of the proposed medical image codec is given in the following subsections.

2.1 Designing of the Image Transformation Block for the Proposed Codec

In general, the compression performance of transform coders highly governed by the energy compaction capability of the image transform utilized in the coder. Over the years, The DCT and the DWT are the most popular transform to facilitate the energy compaction for the transform coders. However, the energy compaction characteristics of the transforms highly depend on the nature of the input image to be compressed. Recently, in [8], the authors have investigated the compression capability of newly proposed DOCST transform on the two different compression modes over various bpp levels. It has been reported that the DOCST transform offers better quality compression than the DCT and DWT in the complete image compression mode, while delivers comparable performance as DCT in 8×8 block-wise compression mode. Moreover, it has been also reported that the higher time complexity of the DOCST transform is a crucial problem related to its real-time application feasibility in 8×8 block-wise compression mode.

These previous findings related to the DOCST transform lead to an important trade-off between the complete image mode and the 8×8 block-wise compression mode. In the complete image mode, the DOCST provides better compression performance than DCT and DWT but offers higher computational complexity. Whereas, in the 8×8 block-wise compression mode the DOCST offers less computational complexity but at the cost of higher time consumption. Therefore, to obtain an intermediate solution for satisfying both the constraints related to the DOCST transform, in this paper, first, a comparative coding gain analysis for the DCT and DOCST transforms is presented for different block sizes ranging from 4×4 to 64×64 .

The coding gain G_{TC} is one of the measures for energy compaction of transforms, defined as the ratio between the arithmetic mean to the geometric mean of the variances σ_i 's (for $i = 1, 2, \dots, N$) of all the components in the transformed vector of length N .

$$G_{TC} = \frac{1}{N} \sum_{i=0}^{N-1} \sigma_i^2 \bigg/ \left[\prod_{i=0}^{N-1} \sigma_i^2 \right]^{\frac{1}{N}} \quad (2)$$

To analyze the coding gain for the DCT and DOCST transforms, three grayscale test images Lena, CT_scan_1, and MRI_1, each of size 512×512 has been used which are shown in Fig. 1. The resultant coding gain comparison of the DCT and DOCST transforms for the three test images on different block sizes is shown in Fig. 2.

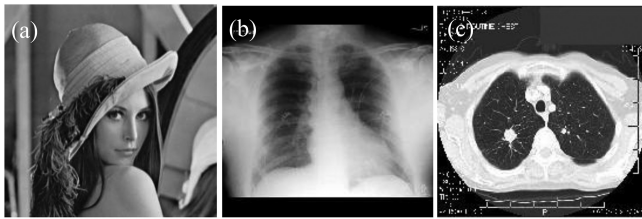


Fig. 1. Test images used for coding gain comparison, (a) Lena, (b) CT_scan_1, (c) MRI_1.

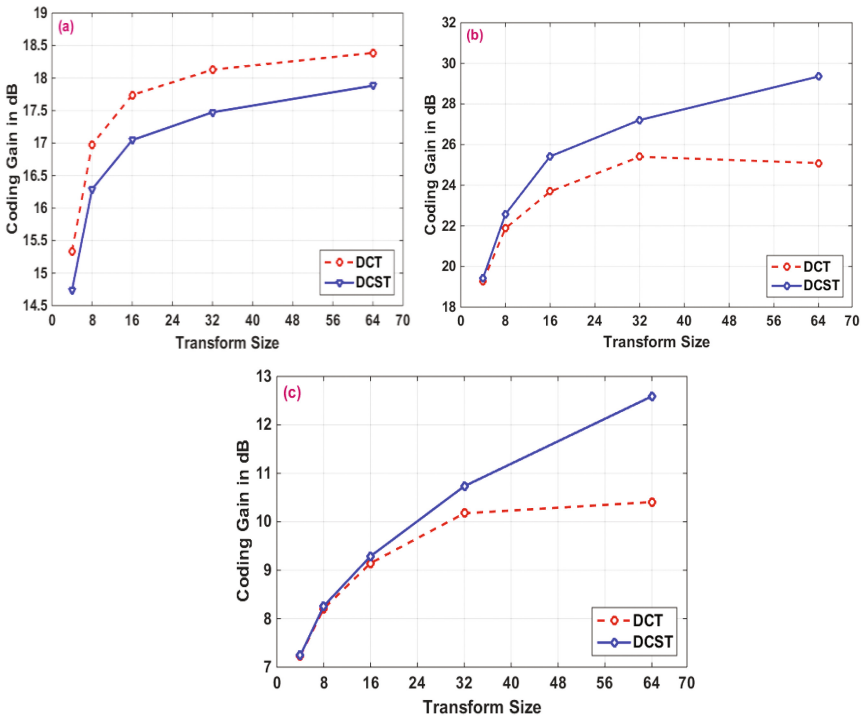


Fig. 2. Coding gain comparison on different block size DCT and DOCST transforms for, (a) Lena image, (b) CT_scan_1 image, (c) MRI_1 image.

From Fig. 2(a), it is evident that for the standard test image Lena the energy compaction capability of the DCT transform is higher than the DOCST transform. Whereas, Fig. 2(b) and (c) depicts that, for both the medical test images of different modalities the energy compaction capability of DOCST transform is much higher than the DCT for block sizes $\geq 32 \times 32$. Hence, to satisfy the computational complexity and higher time consumption constraints of DOCST, this paper proposes the DOCST transform for medical image transformation in a block size of 32×32 instead of size 8×8 . The proposed image transformation block with the DOCST transform is shown in Fig. 3.

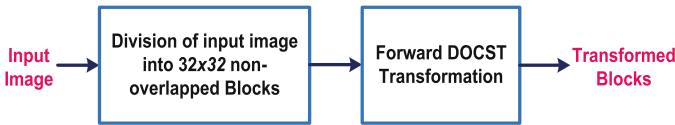


Fig. 3. Proposed image transformation block

The forward DOCST transform can be easily analyzed with the help of Fourier-based Discrete Orthogonal Stock-well Transform (DOST) given in a matrix form as,

$$\text{DOST} = \left(\bigoplus_{i=1}^k D_i \right) \text{DFT} \tag{3}$$

$$D_i = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & (-1)^n \end{bmatrix} F^{-1} \tag{4}$$

where F is the Fourier transform and the direct sum of the matrices D_i forms a block-diagonal matrix with each sub-block (D_i) being an altered inverse Fourier transform. This factorization of the DOST allows to achieve $O(N \log N)$ running times using the Fast Fourier transform.

Further, the DOST can be modified by using a Discrete Cosine Transform (DCT) to generate real-valued transform coefficients. Hence, a cosine version of DOST based on DCT (DOCST) may be defined by simply replacing the DFT in Eqs. (3) and (4) with a DCT:

$$\text{DOCST} = \left(\bigoplus_{i=1}^k \text{DCT}_{ni}^{-1} \right) \text{DCT} \tag{5}$$

When the DCT is used, all frequencies are positive, as a result, higher frequencies are required, so the partitioning in frequency space can be adjusted. The most straightforward choice is to continue using the dyadic partitioning. Given a signal of length 2^N , the widths of the frequency partitions can be defined as follows,

$$n_1 = 1 \text{ and } n_i = 2^{i-2}, \quad 2 \leq i \leq N - 1 \tag{6}$$

2.2 Optimal Bit Allocated Quantization Block for the Proposed Codec

Practically, the transform coefficients are a function of the activities of the image to be compressed. Most often, the available image codecs utilize simple uniform scalar quantization to achieve compression regardless of the activity of the image. Therefore, it is important to design the quantizers with a different number of bits of quantization instead to use the same number of bits for all the quantizers as in the case of existing coders. The average bit rate in bits per pixel (bpp) is usually fixed for a given encoder. This opens the question of how many bits to assign to each quantizer for the transform coefficients to meet the bit budget. This question is generally answered by the optimal bit allocation rule which is defined as,

Statement of the Problem: Given a unitary transform matrix \mathbf{A} of size $N \times N$ and an overall average bitrate of R bpp, the goal is to determine the individual quantizer bits $R_i, 1 \leq i \leq N^2$, such that the mean square error (MSE) between the input and reconstructed images is a minimum. Then the average bit rate of the coder is related to the individual quantizer bits by,

$$R = \frac{1}{M} \sum_{i=1}^M R_i, \quad M = N^2. \tag{7}$$

The optimal bit allocation process is, therefore, an optimization problem which requires satisfying both the rate and distortion criteria's simultaneously. In most of the cases, the available optimization rules do not guarantee either positive or integer value for the number of bits. Therefore, one can use an iterative procedure for the assignment of a positive integer number of bits for quantization. The procedure utilized for the optimum integer bit allocation for the proposed image codec is as follows.

Given a 32×32 block DOCST transform, variances of the transform coefficients σ_i^2 and an average bit budget of R bpp, where $R_t = MR$ is the total number of bits for each of the 32×32 block of quantizers and $M = 32^2$,

Procedure for Integer Bit Allocation [10]:

- (1) Set step j to zero and all quantizer bits to zero, $R_i^{(j)} = 0; 1 \leq i \leq M$.
- (2) Sort the coefficient variances and denote the maximum variance by σ_m^2 .

(3) $R_m^{(j)} \leftarrow R_m^{(j-1)} + 1$ and $\sigma_m^2 \leftarrow \sigma_m^2/2$.

(4) Set $R_t \leftarrow R_t - 1$. If $R_t = 0$, stop. Otherwise, set $j \leftarrow j + 1$ and go to step 2.

Following the above steps, the optimized number of bits for each quantizer of the proposed image codec has been determined which is then mapped to a quantization table of size 32×32 using the following equation to achieve actual quantization of the transform coefficients [11].

$$Q(k) = \frac{(8 \times 27)}{(2b_k - 1)} \quad (8)$$

where, b_k is the average number of bits for the k^{th} quantizer and $Q(k)$ is the quantizer step size for the k^{th} set of transform coefficients. Finally, with the successful designing of all the proposed blocks, the complete block diagram of the proposed medical image encoder is shown in Fig. 4. The proposed decoder exactly reverses the steps of the proposed encoder to determine the reconstructed medical image.

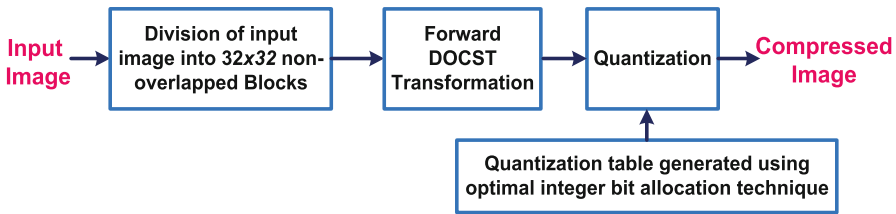


Fig. 4. Block diagram of the proposed medical image encoder.

3 Results and Discussions

This section presents the extensive experimental results for the image compression performance of the proposed medical image codec and its comparative analysis on different bpp levels against the baseline JPEG and JPEG2k codecs [12]. Meanwhile, to particularly analyze the effect of the image transformation and quantization blocks the entropy coding block from both the baseline JPEG and JPEG2k codecs have been removed and final bitrate is again estimated by the first order entropy given in Eq. (1). Finally, the performances of all the codecs are analyzed based on visual quality assessment and image quality index Peak Signal to Noise Ratio (PSNR). All the codecs are tested for the six medical test images of two different modalities such as CT scan and MRI, each of size 512×512 (see Fig. 5).

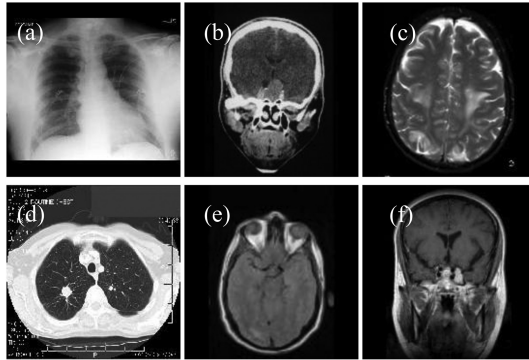


Fig. 5. Test images used for performance evaluation of PMIC against baseline JPEG and JPEG2k codecs, (a) CT_scan_1, (b) CT_scan_2, (c) CT_scan_3, (d) MRI_1, (e) MRI_2, (f) MRI_3.

Further, to present the visual quality assessment the reconstructed images obtained for the first test images CT_scan_1 and MRI_1 from both the modalities for all the coders on 0.097 and 0.12 bpp levels are shown in Figs. 6 and 7 respectively.

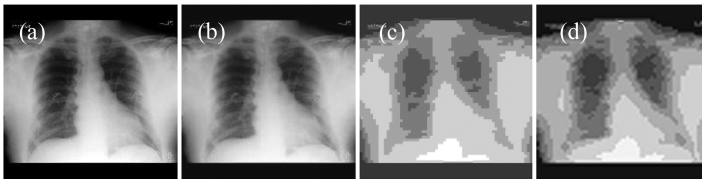


Fig. 6. Observations on 0.097 bpp level compression for the test image CT_scan_1. (a) the original image; reconstructed images for (b) proposed medical image codec, (c) JPEG, and (d) JPEG2k.

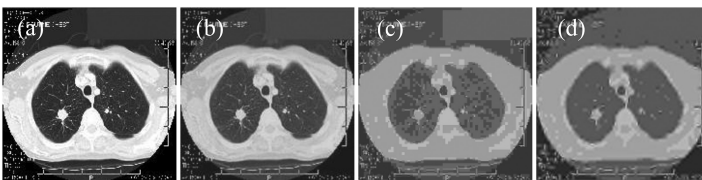


Fig. 7. Observations on 0.12 bpp level compression for the MRI test image MRI_1. (a) the original image; reconstructed images for (b) proposed medical image codec, (c) JPEG, and (d) JPEG2k.

From Figs. 6 and 7, it can be easily observed that the reconstruction quality of the existing JPEG and JPEG2k image codecs are very poor at the lower bpp levels for both

the test images; whereas on the same compression level the proposed codec delivers the very good visual quality of the reconstructed images. Therefore, the proposed codec offers significant quality improvement as compared to the existing codecs and hence able to compress the medical images on higher compression levels with the best quality. Further, a comparative rate-distortion performance of all the codecs based on image quality index PSNR for all the six medical test images is shown in Fig. 8.

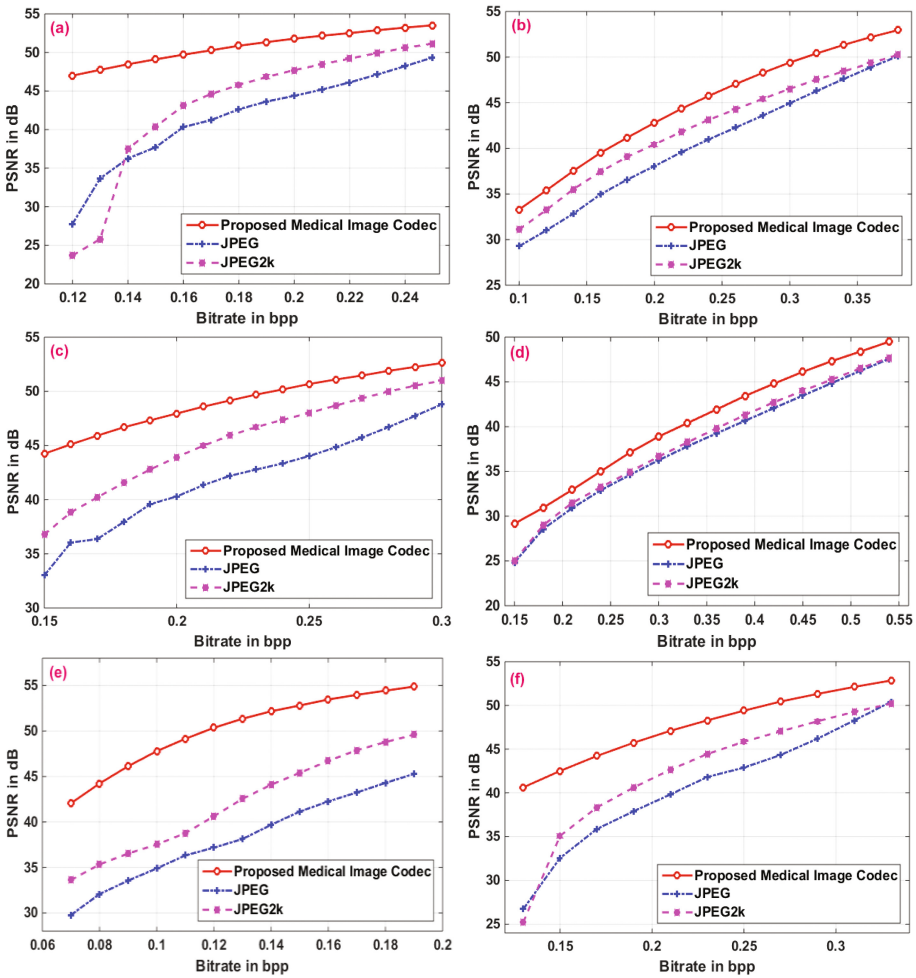


Fig. 8. Rate-distortion performance of proposed medical image codec against the existing JPEG and JPEG2k codecs for, (a) CT_scan_1 image, (b) CT_scan_2 image, (c) CT_scan_3 image, (d) MRI_1 image, (e) MRI_2 image, (f) MRI_3 image.

From Fig. 8, it can be easily observed that the proposed medical image coder delivers significantly higher PSNR for all the six test images on all the tested bpp

levels. The visual quality assessment and the comparative rate-distortion performance presented in this section clearly validates that the developed codec outperforms the existing codecs for medical image compression with significant quality improvement.

3.1 Time Complexity Analysis of the Proposed Medical Image Codec

The encoding and decoding time consumption which is also known as time complexity of the image coders plays an important role in the real-time compression applications. Therefore, this subsection presents a comparative time complexity analysis for the proposed codec to validate its real-time application feasibility against the existing JPEG and JPEG2k codecs. For the time complexity evaluation of the image codecs, the total time required for the encoding and reconstruction process has been obtained over various bpp levels for all the six test images. Subsequently, the average values of the respective time requirements have been taken for the comparison and analysis. The obtained values of the Average Time Requirements (ATR) for the proposed medical image codec, JPEG, and JPEG2k codecs are listed in Table 1.

Table 1. Time complexity comparison of image codecs

Test Image	Average Time Requirements (ATR) for image codecs in seconds		
	Proposed medical image codec	JPEG codec	JPEG2k codec
CT_scan_1	1.6626	3.8988	0.1432
CT_scan_2	1.6764	3.6994	0.1398
CT_scan_3	1.5694	3.7104	0.1482
MRI_1	1.6264	3.8785	0.1476
MRI_2	1.5768	3.6551	0.1371
MRI_3	1.6596	3.8681	0.1459

From Table 1, it is clear that the proposed medical image codec takes average time consumption of about 1.62 s to encode and reconstruct the input image which is slightly higher than the JPEG2k and less than 50% of the popular JPEG codec. Therefore, the proposed codec is fast as compared to the existing JPEG codec and hence highly suitable for the real-time medical image compression applications.

4 Conclusions

In the present paper, an advanced image codec has been developed to deliver high-quality medical image compression, especially over lower bpp levels. The ultimate aim was to support the telemedicine and tediagnosis process by enhancing the transmission of higher dimension medical images through the low bandwidth channel with higher quality. To justify and validate the superiority of the proposed medical image codec, it has been extensively evaluated and compared with the existing JPEG

and JPEG2k codecs on the higher compression levels. The obtained result shows that the proposed codec offers a very high gain in the visual quality and the PSNR index as compared the state of the art medical image codecs.

References

1. Huang, H.K.: PACS and Imaging Informatics: Basic Principles and Applications. Wiley, New York (2010)
2. Jayaraman, S., Esakkirajan, S., Veerakumar, T.: Digital Image Processing, 1st edn. Tata McGraw Hill Education, New Delhi (2009)
3. Wallace, G.K.: The JPEG still picture compression standard. *Commun. ACM* **34**, 30–44 (1991)
4. Thakur, V.S., Thakur, K.: Design and implementation of a highly efficient grey image compression codec using fuzzy based soft hybrid JPEG standard. In: International Conference on Electronic Systems, Signal Processing and Computing Technologies (ICESC), pp. 484–489, January 2014
5. Thakur, V.S., Dewangan, N.K., Thakur, K.: A highly efficient grey image compression codec using neuro-fuzzy based soft hybrid JPEG standard. In: 2nd International Conference on Emerging Research in Computing, Information, Communication and Applications (ERCICA), vol. 1, pp. 625–631 (2014)
6. Thakur, V.S., Gupta, S., Thakur, K.: Optimum global thresholding based variable block size DCT coding for efficient image compression. *Biomed. Pharmacol. J.* **8**(1), 453–468 (2015)
7. Christopoulos, C., Skodras, A., Ebrahimi, T.: The JPEG2000 still image coding system: an overview. *IEEE Trans. Consum. Electron.* **46**, 1103–1127 (2000)
8. Thakur, V.S., Gupta, S., Thakur, K.: Perceptive performance analysis of Discrete Orthogonal Cosine Stockwell Transform for low bit-rate image compression. In: International Conference on Recent Trends in Engineering, Science and Technology (ICRTEST 2016), no. 2, pp. 251–257 (2016)
9. Ladan, J., Vrscaj, E.R.: The Discrete Orthonormal Stockwell Transform and Variations, with applications to image compression. In: Kamel, M., Campilho, A. (eds.) ICIAR 2013. LNCS, vol. 7950, pp. 235–244. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39094-4_27
10. Sayood, K.: Introduction to Data Compression. Morgan Kaufman, San Francisco (1996)
11. Fung, H.T., Ranker, K.J.: Design of image-adaptive quantization tables for JPEG. *J. Electron. Imaging* **4**(2), 144–150 (1995)
12. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital Image Processing Using MATLAB, 2nd edn. Prentice-Hall, Upper Saddle River (2010)

Coprime Mapping Transformation for Protected and Revocable Fingerprint Template Generation

Rudresh Dwivedi^(✉) and Somnath Dey

Discipline of Computer Science and Engineering, Indian Institute of Technology
Indore, Indore 453446, India
{phd1301201006,somnathd}@iiti.ac.in

Abstract. Compromise of biometric data may cause permanent loss of identity since the biometric information is intrinsically linked with the user. To revoke the stolen biometric template, the concept of cancelable biometrics has been introduced. The idea behind cancelable biometric is to transform the original biometric template into a new template and perform matching in the transformed domain. In this paper, a coprime transformation scheme has been proposed to generate the cancelable fingerprint template. The method divides the fingerprint region into a number of sectors with respect to each minutiae point and identifies the nearest-neighbor minutiae in each sector. Then, ridge-based features for all minutiae points are computed and mapped onto co-prime positions of a random matrix to generate the cancelable template. The proposed approach achieves an EER of 1.82, 1.39, 4.02 and 5.77 on DB1, DB2, DB3 and DB4 datasets of the FVC2002 database, respectively. Experimental results indicate that the method outperforms in comparison to the current state-of-the-art. Moreover, the proposed method fulfills the necessary requirements of diversity, revocability, and non-invertibility with a minor performance degradation caused by the transformation.

Keywords: Biometric · Fingerprint verification · Template protection

1 Introduction

1.1 Background

Compromising the stored biometric template causes permanent losing his/her identity due to irreplaceable and irrevocable characteristics of original biometric data. There are several privacy issues associated with the sharing of biometric information across many applications [1]. Therefore, it is necessary to provide biometric template protection. The concept of cancelable biometric has been introduced for template protection which state that a transformed template is required to be stored instead of the original biometric template. The transformation relies on an irreversible function such that it is difficult to discover the original template even if the attacker discovers the transformation function and the transformed template. In the case of compromise, a new template can be

derived by altering the parameter values of the transformation function. Further, it should not exhibit significant performance degradation in comparison to the true biometric system.

1.2 Existing Approaches

Recently, various approaches for cancelable template design have been proposed in the literature. Ratha et al. [1] introduced cartesian, polar, and functional transformation for fingerprint template security. Das et al. [2] introduced a graph structure based on the nearest-neighbor distance from core point to all other minutiae points. However, the methods proposed by Ratha [1] and Das et al. [2] require core point to align two fingerprints before transformation. However, the detection of the core point is not always possible.

Lee et al. [3] proposed a method to map aligned minutiae points into a 3-D array based on the minutiae orientation and difference between minutiae coordinates. The array is visited in sequence to derive a bit-string which is permuted based on a user-specific PIN and the type of minutiae. In the alignment-free method proposed by Wang et al. [4], pair-minutiae vectors are quantized, indexed and converted to bit-string. Then, a user specific PIN is applied to the complex vector derived by taking discrete Fourier transform onto bit-string. Moujahdi et al. [5] proposed fingerprint shell which utilizes the distance between the singular point and all other minutiae points. The distances with an addition of user-specific key are sorted in ascending order to derive spiral curve. In another work, Wang et al. [6] presented a way to protect the bit-string derived using the method proposed in [4]. The bit-string is utilized as an input to FIR filter with a user-specific key. The performance of the methods proposed in [3–6] degrades if user-specific token is compromised. Further, Wang et al. [7] proposed a method which utilizes the partial Hadamard transform to the derived bit string.

Cappelli et al. [8] proposed a novel minutiae representation MCC (Minutiae cylinder Code) which constructs a 3-D cylindrical structure around each minutiae neighborhood. Later, Ferrara et al. [9] proposed protected-MCC (P-MCC) which applies binary-KL projection onto MCC templates to overcome security concerns against non-invertibility in MCC [8]. However, further investigations unveil the irrevocability issue of P-MCC. To achieve revocability, Ferrara et al. [10] proposed two-factor protected Minutiae Cylinder-Code (2P-MCC) which performs partial permutation using a secret key over the cylinders in P-MCC.

1.3 Contributions

To alleviate the issues of the existing methods described above, we propose a novel cancelable fingerprint template generation method based on coprime mapping transformation. Our contributions in this work are highlighted in the following:

- (1) Ridge features are evaluated under ridge coordinate system to deal with rotation, scale and translation distortions in the input fingerprint image.

- (2) The proposed work does not rely on pre-alignment of the core or singular points as it is hard to detect the singularities in poor quality fingerprint images.
- (3) The nearest-neighbor transformation is applied around each minutia to derive a fixed length descriptor instead of fixed-radius transformation. This overcomes the limitation of performance degradation caused due to the border minutiae points.
- (4) We have tested our approach with respect to the desirable criteria for cancelable transformation i.e. revocability, irreversibility, and diversity.
- (5) The performance of the proposed method is evaluated on all datasets of FVC2002. The experimental results show that our approach performs better than the existing approaches.

The organization of this paper is as follows. Section 2 describes the proposed scheme for cancelable template generation. Experimental results are demonstrated in Sect. 3. Section 4 presents the security analysis. Concluding remarks and course of future work are described in Sect. 4.

2 Proposed Scheme

The overall design for the proposed method is illustrated by the block diagram shown in Fig. 1. The proposed method consists of three main tasks including pre-processing and minutiae extraction, feature extraction, and cancelable template generation.

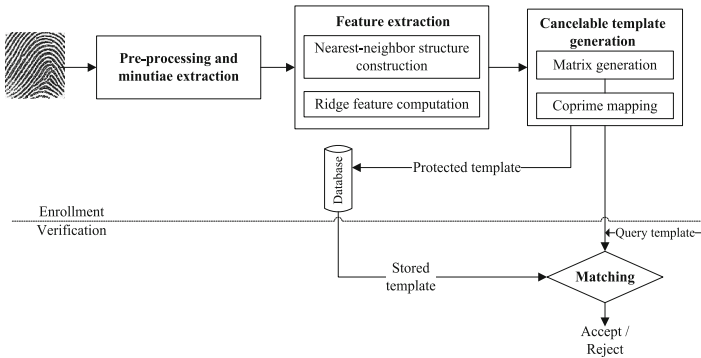


Fig. 1. Block diagram of the proposed method

2.1 Pre-processing and Minutiae Extraction

In this work, pre-processing and minutiae point extraction is performed by the approach described in [11]. The minutia points are represented as:

$$V_{up} = \{m_i\}_{i=1}^n$$

$$m_i = (x_i, y_i, \theta_i) \quad (1)$$

where, V_{up} is a set of unprotected minutiae points derived from a fingerprint image, m_i is the i^{th} minutiae point and n is the total number of minutiae points in V_{up} . The minutiae point m_i is represented by the coordinate (x_i, y_i) and the minutiae orientation θ_i . The preprocessing task also outputs a thinned fingerprint image which is utilized for feature extraction.

2.2 Feature Extraction

There is a necessity to compute transformation invariant features from a fingerprint image since performance could degrade by rotation, translation and scaling transformation caused at the time of acquisition. In this work, we compute ridge features to deal with rotation and scale deformations present in the input fingerprint image. Feature extraction involves two steps: nearest-neighbor structure construction and ridge feature computation.

Nearest-Neighbor Structure Construction: After the preprocessing task, we obtain the thinned output image and minutiae information from the input fingerprint. One of the minutiae from the minutiae set V_{up} is considered as a reference minutiae. Next, we construct the nearest-neighbor structure around the reference minutiae point utilizing ridge coordinate system as shown in Fig. 2(a). The ridge coordinate system allocates the reference axis coinciding with the orientation of the selected minutiae. Further, the fingerprint region is divided into ‘s’ sectors of equal angular displacement utilizing ridge coordinate system as displayed in Fig. 2(a).

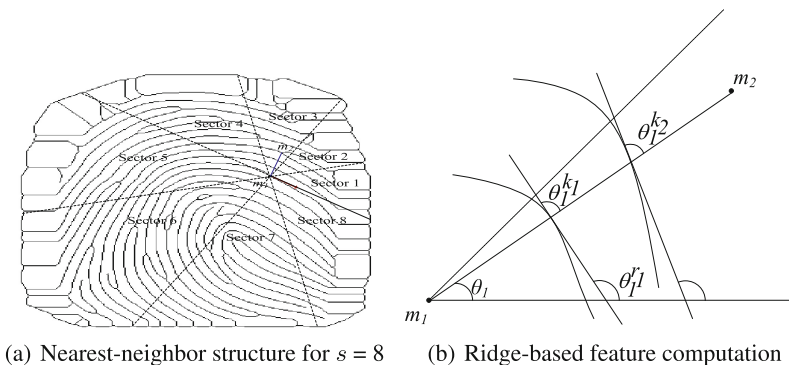


Fig. 2. Feature extraction

Ridge Feature Computation: First, each minutia is considered as a reference. Next, ridge count and average ridge orientation between reference minutiae and nearest minutiae in each sector are calculated. Ridge count is evaluated by counting the ridges between reference minutiae and nearest neighbor minutia. For example, ridge count between two minutiae points (say m_1 and m_2) is 2 as shown in Fig. 2(b). To compute ridge orientation, the angle subtended by the tangent line and the straight line connecting two minutiae points is measured for each ridge crossing. For example, the orientation of the first ridge in the first sector as shown in Fig. 2(b), θ_1^{k1} can be evaluated as:

$$\text{for sector 1: } \theta_1^{k1} = \theta_1^{r1} - \theta_1$$

where θ_1 is the slope of the line connecting reference minutiae and nearest neighbor minutia in the first sector. θ_1^{r1} , is the angle subtended by the tangent line from the intersection point of first ridge and reference axis. Similarly, we can find out the orientation of second ridge, θ_1^{k2} and evaluate the mean ridge orientation for example shown in Fig. 2(b). The mean ridge orientation for each sector can be formulated as defined in Eq. (2).

$$\text{for } i^{th}, \text{ sector: } r_{ori} = \frac{(\theta_i^{r1} - \theta_i) + (\theta_i^{r2} - \theta_i) + \dots + (\theta_i^{rNr_i} - \theta_i)}{Nr_i} \quad (2)$$

where Nr_i is the total number of ridges between the reference and nearest minutiae in the i^{th} sector. We store the ridge features into a 2-D matrix (F). For example, if a fingerprint image contains n minutiae points then, the feature matrix F will contain $n \times 2s$ entries including s ridge count and s average ridge orientation considering s sectors in a fingerprint image. We assign zero to the ridge features corresponding to a sector if no minutia point is located in that sector. At the time of matching, we do not consider the sectors with no minutiae point.

2.3 Cancelable Template Generation

The generation of cancelable fingerprint template involves two tasks: matrix generation and co-prime mapping.

Matrix Generation: We map the feature matrix into a high-dimensional matrix to derive the protected template. For this purpose, a random matrix $CanTemp$ of size $T \times T$ is generated with a seed (ρ). The value of T is equal to $n \times 2s$ where n and s are the total number of minutiae points in the input fingerprint image and the number of sectors around a reference minutiae, respectively.

Co-prime Mapping: We map the feature matrix $F_{n \times 2s}$ into $CanTemp$ such that there will be no overlapping. To perform this, we use co-prime based mapping in our method which maps all elements of F at T places of $CanTemp$.

Rest of the entries of matrix are filled with is filled with some random data. The following four keys are utilized for mapping:

(1) k_1 : initial row position (2) k_2 : initial column position (3) k_3 : number of row jump from initial position (4) k_4 : number of column jump from initial position.

The start position is calculated based on the user-specific key. We start at position (k_1, k_2) in matrix *CanTemp*. The next position (*NP*) is computed based on the row and column jump to the initial position using the following relation described in Eqs. (3) and (4):

$$NP_i = \begin{cases} k_1 + k_3 & \text{if } (k_1 + k_3 \leq T) \\ k_1 + k_3 - T & \text{if } (k_1 + k_3 > T) \end{cases} \tag{3}$$

$$NP_j = \begin{cases} k_2 + k_4 & \text{if } (k_2 + k_4 \leq T) \\ k_2 + k_4 - T & \text{if } (k_2 + k_4 > T) \end{cases} \tag{4}$$

To avoid overlapping in the matrix, the co-prime mapping is adopted. In this technique, we select the value of k_3 and k_4 such that both should be co-prime with T as defined in Eq. (5).

$$\begin{aligned} GCD(k_3, T) &= 1 \quad \forall k_3 \in [2, T] \\ GCD(k_4, T) &= 1 \quad \forall k_4 \in [2, T] \end{aligned} \tag{5}$$

For example, if the key values for start position are $k_1 = 2$ and $k_2 = 2$, respectively and the key values for row and column jump are $k_3 = 3$, $k_4 = 5$, then the co-prime based mapping is shown in Fig. 3.

	1	2	3	4	5	6	7	8
1	1	0	5	9	3	1	10	19
2	6	2	3	7	11	15	28	32
3	0	7	7	0	8	12	11	17
4	9	4	9	4	7	0	17	8
5	6	5	6	4	12	10	45	19
6	4	8	3	11	19	5	27	14
7	2	7	1	0	22	3	34	21
8	8	4	0	6	11	7	19	14

Fig. 3. Example of co-prime based mapping procedure

2.4 Matching

Fingerprint matching refers to the process of comparing an enrolled fingerprint template (say *CT*) and a query fingerprint template (say *QT*) to return a matching score. In our method, matching is performed in two steps: Local matching and global matching.

Local Matching: In local matching, ridge feature set corresponding to a minutiae point from QT is compared with ridge feature set for a minutiae point of CT to return local match score. Mapped ridge feature set in the QT and CT are accessed using user-specific keys k_1 , k_2 , k_3 and k_4 . We compute the Euclidean distance between the mapped non-zero entries of the query and enrolled template. Next, we compute the mean of the minimum distances corresponding to each non-zero entries of the two ridge features sets of CT and QT as described in Eq. (6).

$$e_dist = \sqrt{(QT_N[i][1] - CT_M[j][1])^2 + (QT_N[i][2] - CT_M[j][2])^2} \quad (6)$$

Global Matching: In global matching, we compute the number of matched minutiae points between QT and CT utilizing the local match scores by comparing each ridge-feature set from QT with each ridge-feature set from CT . Next, overall matching score is evaluated by the number of matched minutiae points divided by the number of minutiae points in QT as described in Eq. (7).

$$overall_match_score = \frac{match_minutiae_count}{N} \quad (7)$$

3 Experimental Results and Analysis

In our experiment, we use four datasets DB1, DB2, DB3 and DB4 of FVC2002 database [12] since the most of the existing approaches utilized these datasets. Each datasets DB1, DB2, DB3 and DB4 of FVC2002 contains a total of 800 images of 100 subjects with eight samples each. The performance of the method is evaluated with four parameters: False Acceptance Rate (FAR), False Rejection Rate (FRR), Equal Error Rate (EER) which is defined as the error rate when the FRR and FAR holds equality, and GAR is computed as 1-FRR.

3.1 Validation of Parameter: Number of Sectors (s)

After the preprocessing steps, the proposed method divides the input fingerprint image into the s number of sectors with equal angular width. To validate the parameter s , we have performed a number of experiments considering distinct angular widths. We have computed the EER with angular width of 15° , 30° , ... and 90° corresponding to $s = 24, 12, \dots$ and 4, respectively. The performance for the different number of sectors is reported in Table 1. It has been observed that the method performs the best for $s = 8$ on each of the datasets of FVC2002. It has also been observed that for high values of s , EER increases as there are more number of sectors without minutiae points. Therefore, we have considered $s = 8$ for all other experiments.

Table 1. EER obtained for databases FVC 2002 DB1, DB2, DB3 and DB4 in same key scenario

Number of sectors (s)	EER (in %)			
	FVC2002 DB1	FVC2002 DB2	FVC2002 DB3	FVC2002 DB4
4	3.93	3.79	5.86	6.83
8	1.82	1.39	4.02	5.77
16	5.04	4.93	8.83	12.7
32	9.63	5.19	11.24	19.3

3.2 Performance

We have followed FVC protocol to evaluate our method which compares each subject against the first sample of the remaining subjects to calculate impostor scores and each sample is compared against the remaining samples of the same subject to calculate the genuine score. Therefore, 4950 and 2800 impostor and genuine comparisons are required respectively if all samples are enrolled for each set of the FVC2002 database. Further, we have conducted the experiments under two scenarios to evaluate the performance of our method: Same key scenario and different key scenario.

Same Key Scenario: In this scenario, we assume that a user’s key is stolen. In this case, an imposter utilizes the key as a genuine user to gain access into the system. To rectify this attack, we apply same keys (i.e. k_1 , k_2 , k_3 and k_4) to enroll all users. The proposed method is applied onto DB1, DB2, DB3 and DB4 dataset of database FVC2002. Figure 4 represents the ROC curves for each dataset of FVC2002 for the optimal value of parameter s (i.e. $s = 8$).

For FVC2002 database, we achieve an EER of 1.82, 1.39, 4.02, and 5.77 for DB1, DB2, DB3, and DB4, respectively using FVC protocol. Out of all FVC2002 datasets, the method performs better on DB1 and DB2 as these datasets contain more number of good quality images as compared to datasets DB3 and DB4. Further, the dataset DB3 and DB4 contain less number of minutiae points per image due to poor quality images as compared to dataset DB1 and DB2. As a result, we achieve high EER for DB3 and DB4 datasets.

Different Key Scenario: To test our method in different key scenario, we use different keys (i.e. k_1 , k_2 , k_3 and k_4) to enroll different users. We obtain an EER of 0 for DB1, DB2 and DB4 datasets and an EER of 0.09 for the dataset DB3 of FVC2002. Therefore, it is evident that our approach performs better in the different key scenario.

3.3 Baseline Comparison

For baseline comparison, we perform two set of experiments. In the first experiment, we compute the EER using the original fingerprint template comprising

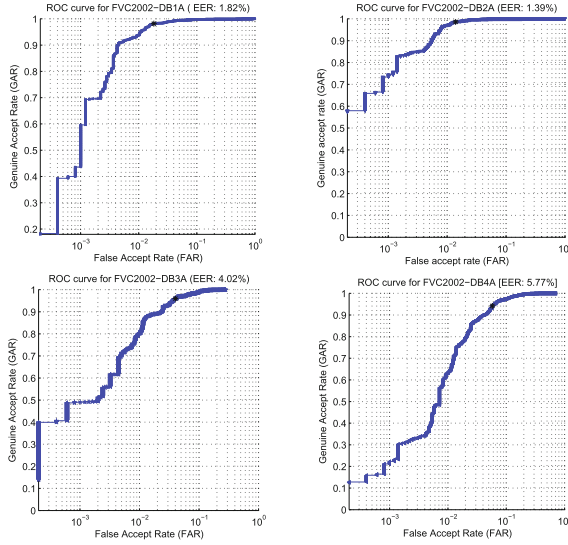


Fig. 4. ROC curves for FVC 2002 DB1, DB2, DB3 and DB4 under same key scenario

ridge features. In the second experiment, we apply coprime mapping transformation utilizing the keys (k_1 , k_2 , k_3 and k_4) and evaluate the performance. Table 2 shows that the performance is degraded by 0.19%, 0.41%, 0.05%, and 0.39% for DB1, DB2, DB3, and DB4 dataset of FVC2002 database, respectively. From the reported results, it is evident that the performance degradation caused by the transformation is very low.

Table 2. Baseline comparison for FVC2002 database

FVC2002	EER (in %)		Performance degradation
	Without cancelable transformation	With cancelable transformation	
DB1	1.47	1.82	0.19
DB2	0.89	1.39	0.41
DB3	3.81	4.02	0.05
DB4	3.49	5.77	0.39

3.4 Comparison with Existing Approaches

The proposed method is compared with methods [2–7, 9, 10] described in Sect. 1. Table 3 shows the comparison in terms of EER. From Table 3, it has been observed that the proposed method performs better as compared to the

approaches proposed in [2,3,5,10]. However, the performance of our method is slightly lower than the approach in [9,13] yet comparable to the existing template protection approaches. Therefore, from the reported results it is evident that our approach outperforms over the existing methods.

Table 3. EER obtained for databases FVC 2002 DB1, DB2, DB3 and DB4 in same key scenario

Methods	EER (in %)			
	FVC2002 DB1	FVC2002 DB2	FVC2002 DB3	FVC2002 DB4
Das et al. [2]	2.27	3.79	-	-
Moujahdhi et al. [5]	4.28	1.45	-	-
Wang et al. [4]	3.5	5	7.5	-
Lee et al. [3]	10.3	9.5	6.8	-
Wang et al. [6]	3	2	7	-
Wang et al. [7]	1	2	5.2	-
Ferrara et al. [9]	1.88	0.99	5.24	4.84
Ferrara et al. [10]	3.3	1.8	7.8	6.6
Proposed method	1.82	1.39	4.02	5.77

‘-’ indicates that the author(s) have not reported the results or results are reported for the partial dataset, in their work.

4 Security Analysis

A cancelable biometrics system needs to satisfy the criteria of irreversibility, revocability, and diversity as described in Sect. 1. In the following subsections, we will analyze our method with respect to these criteria.

4.1 Irreversibility Analysis

To analyze the irreversibility, we assume that an adversary is able to reveal the stored protected template *CanTemp*. In this case, the attacker cannot be able to reveal original template (F) as he does not have any information about the four keys utilized for mapping. For example, if the fingerprint image contains 50 minutiae points and it is divided into 8 sectors then the original template (F) and protected template *CanTemp* would contain 800 cells and 640000 cells, respectively. It is very hard to compute initial positions (k_1, k_2) and next positions (k_3, k_4) to retrieve the entries of original template as there are $640000 \times 640000 = 409$ billion brute force attempts are required.

Further, if the attacker reveals the keys (k_1, k_2, k_3 and k_4) utilized for mapping. In this case, the attacker cannot be able to derive original template since keys k_1, k_2, k_3 and k_4 comprise of random coprime entries. From random coprime entries, it is impossible to retrieve any information about the original template.

4.2 Revocability Analysis

The revocability states that a new template must be issued if a stored protected template is compromised. To test the revocability of the method, we derived 100 different transformed templates by varying the parameter values from the same fingerprint. Next, genuine, imposter and Pseudo-imposter distribution are calculated for the FVC2002-DB1 dataset. From the experiment, we achieve 0% average FAR. The mean and standard deviation ($\mu; \sigma$) of genuine, imposter and pseudo-imposter are 0.3931; 0.019, 0.9231; 0.0326, and 0.891; 0.0376, respectively. From the computed distribution, we observe that there is a strong overlap between the pseudo-imposter and imposter distributions as shown in Fig. 5. This implies that the templates derived with different keys from the same subject are different enough to prevent the cross-matching attack. Therefore, it can be stated that the transformed template differs from the compromised template although derived from the same fingerprint.

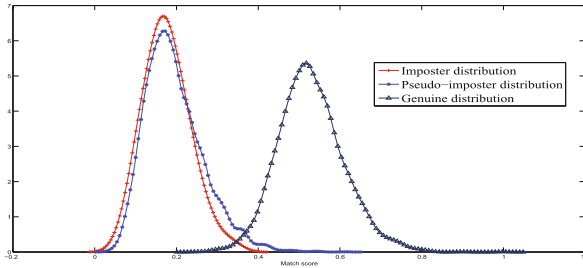


Fig. 5. Genuine, imposter and pseudo-imposter distribution for FVC2002 DB2

4.3 Diversity Analysis

It is essential that it should derive numerous templates without allowing cross-matching over various applications. Numerous different templates can be achieved by altering the key values (k_1, k_2, k_3, k_4) and seed value (ρ). Moreover, a change in the number of sectors (s) also suffices the generation of numerous templates.

5 Conclusion

In this paper, we have proposed a novel cancelable fingerprint template generation technique. The proposed technique does not depend on detection of singularities (core/delta). In this method, the input fingerprint image is divided into a number of sectors of equal angular partition. Invariant ridge features for the nearest neighbor minutiae in each sector are computed considering each minutia as a reference. Further, ridge features are mapped into a higher dimension

random matrix in coprime manner to derive the protected template. Experiments carried out over DB1, DB2, DB3 and DB4 datasets of FVC2002 database show a significant performance improvement as compared to the current state-of-the-art. Further, the security analysis ensures that our approach fulfills the necessary criteria for template protection schemes preserving the recognition accuracy. However, the computation of ridge feature for low-quality fingerprint and partial fingerprint images is a challenging task. This would be our future research direction.

Acknowledgment. The authors are thankful to SERB (ECR/2017/000027), Deptt. of science & Technology, Govt. of India for providing financial support to carry out this research work. Also, we would like to thank Kamal Meena and Rajesh Verma for coordinating and working with us.

References

1. Ratha, N.K., Chikkerur, S., Connell, J.H., Bolle, R.M.: Generating cancelable fingerprint templates. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(4), 561–572 (2007)
2. Das, P., Karthik, K., Chandra Garai, B.: A robust alignment-free fingerprint hashing algorithm based on minimum distance graphs. *Pattern Recogn.* **45**(9), 3373–3388 (2012)
3. Lee, C., Kim, J.: Cancelable fingerprint templates using minutiae-based bit-strings. *J. Netw. Comput. Appl.* **33**(3), 236–246 (2010)
4. Wang, S., Hu, J.: Alignment-free cancelable fingerprint template design: a densely infinite-to-one mapping approach. *Pattern Recogn.* **45**(12), 4129–4137 (2012)
5. Moujahdi, C., Bebis, G., Ghouzali, S., Rziza, M.: Fingerprint shell: secure representation of fingerprint template. *Pattern Recogn. Lett.* **45**, 189–196 (2014)
6. Wang, S., Hu, J.: A blind system identification approach to cancelable fingerprint templates. *Pattern Recogn.* **54**, 14–22 (2016)
7. Wang, S., Deng, G., Hu, J.: A partial Hadamard transform approach to the design of cancelable fingerprint templates containing binary biometric representations. *Pattern Recogn.* **61**, 447–458 (2017)
8. Cappelli, R., Ferrara, M., Maltoni, D.: Minutia cylinder-code: a new representation and matching technique for fingerprint recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(12), 2128–2141 (2010)
9. Ferrara, M., Maltoni, D., Cappelli, R.: Noninvertible minutia cylinder-code representation. *IEEE Trans. Inf. Forensics Secur.* **7**(6), 1727–1737 (2012)
10. Ferrara, M., Maltoni, D., Cappelli, R.: A two-factor protection scheme for MCC fingerprint templates. In: *International Conference of the Biometrics Special Interest Group*, pp. 1–8 (2014)
11. Abraham, J., Gao, J., Kwan, P.: *Fingerprint matching using a hybrid shape and orientation descriptor*. INTECH Open Access Publisher (2011)
12. Fingerprint Verification Competition: FVC 2002 database. <http://bias.csr.unibo.it/fvc2002/databases.asp>
13. Boulton, T.E., Scheirer, W.J., Woodworth, R.: Revocable fingerprint biotokens: accuracy and security analysis. In: *IEEE International Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 1–8, June 2007

Supervised Asymmetric Metric Extraction: An Approach to Combine Distances

Archil Maysuradze¹, B.H. Shekar²(✉), and Mikhail Suvorov³

¹ Lomonosov Moscow State University, Moscow, Russia
maysuradze@cs.msu.ru

² Mangalore University, Mangalore, Karnataka, India
bhshekar@gmail.com

³ CitySoft LLC, Moscow, Russia
severe013@gmail.com

Abstract. We propose a novel supervised distance metric extraction technique. Given several original metrics and a finite set of labeled objects, the problem is to produce a new metric which better agrees with the labels of the training objects. The problem may be seen as the best single metric extraction from a metric-based description. Feature-based object descriptions are not used even implicitly. Unlike many metric approaches, we treat intraclass and interclass distances differently. The metric extraction problem is reduced to a linear programming problem that makes it possible to use effective optimization techniques. It is proved that an admissible solution always exists and hence there is no need to introduce any soft-constraint extension and the number of variables remains small. Thus, the computational complexity depends mainly on the original metric calculation. The method is empirically tested on biometric data where all the original and derived metrics are calculated in real time.

Keywords: Multiple distance metrics · Similarity measures
Metric-based descriptions · Dimensionality reduction
Combined classifier · Constrained optimization

1 Introduction

In machine learning theory and data mining applications, there are increasingly frequent situations when different ways to measure similarity are set on the same objects. Such situations are typical in information retrieval, computer vision, biology, social systems, finance, etc. In many of these domains, similarity engineering is an important way to incorporate expert knowledge and similarity learning is a way to produce a similarity function on the basis of some constraints. In mathematics, one way to formalize the notion of similarity is a metric, i.e. a distance function meeting metric requirements. In machine learning and data mining, researchers usually admit zero distances between non-coinciding objects.

Thus, it would be formally correct to talk about semimetrics, or pseudometrics, however, for simplicity of reading, we will use imprecise word metric. A set possessing several metrics is called a multimetric space. When several metrics are set on the object space, we say that there is a metric-based description.

By analogy with conventional machine learning problems where an object is described by several features, it is reasonable to expect that when we have several metrics even if the individual metrics do not discriminate between classes accurately their combination will improve the quality of the discrimination. Therefore, we recognize a notion of metric dimension reduction methods which produce a set of derived metrics from a set of original metrics.

For the fundamental problems of machine learning primarily supervised learning tasks some metric models have already become classic, especially the methods of the nearest neighbors, Parzen-Rosenblatt windows, and potential functions. It is important to note that these classical methods are formulated, theoretically justified, and studied for situations with one metric only. Therefore, it is necessary to recognize the problem of aggregating original multimetric information, i.e. it is required to construct a new metric on the whole population from a set of distances on the same objects. Like in the case of feature-based descriptions, when formalizing the task of metric-based description aggregation, one can proceed in different ways. Some metric analogs of blind signal separation methods were considered earlier, for example the metric versions of principal component analysis and nonnegative matrix factorization are found in [1]. Those methods are unsupervised.

In this paper, we consider the case where the original distances are computed on a finite training sample, i.e. for all the pairs of the objects of a finite set of samples. Actual class labels are given for the training objects. It is required to construct a new metric on the whole population (not only training objects). The derived metric must optimally (in a sense) agree with the labels of the training objects. This task can be attributed to the group of metric extraction tasks. From the numerous well-known metric learning tasks, our task differs in that a new metric is constructed from the original metrics, and not from features of any kind: no individual objects descriptions are given. We propose an approach in which the problem is reduced to the problem of linear programming. The theoretical properties of the problem are provided along with empirical results to exhibit the performance of the proposed approach.

2 Metric Approaches in Machine Learning

In this work, we distinguish between metric learning and metric extraction. By analogy with the feature extraction which transforms several original features to several derived ones, we say that metric extraction produces new metrics given original metrics. Metric extraction is a kind of dimensionality reduction of metric-based descriptions. Unlike metric extraction, metric learning is a task of producing metrics from conventional feature-based object descriptions. In the

paper, metric extraction ignores object features (except class labels). The original metrics may, but need not be based on object features. The original metrics can be hand-crafted or produced by metric learning.

There is a far more general notion of multiple distance combination. It is not uncommon in machine learning applications to use several distances. When there is no single best distance in a domain and each distance has its own advantages and limitations, the idea to combine the distances immediately comes to mind. And there are many different strategies to do so, metric extraction being just a group of them. Much more often than metric extraction, applied researchers use ensemble approaches to combine single-metric classifiers.

Both metric learning and metric extraction can have different levels of supervision. In the paper, we consider a supervised metric extraction problem. Conventional supervised learning has the common form of training: each training object is annotated with its class label. Training information for supervised metric approaches may be specified in different forms. The two most common forms are link constraints and relative constraints. In the paper, we stick to link constraints: there is a set of must-link pairs of training objects and a set of cannot-link pairs of training objects. In the conventional case when each training object is annotated with its class label, the must-link set contains all the pairs of the same class training objects and the cannot-link set contains all the pairs of training objects from different classes. Semantically, the must-link and cannot-link sets jointly are expected to satisfy the equivalence relation axioms. Formally, we will not require such constraints in this work though in most applications the equivalence relation requirements are met. The general informal objective of both supervised metric learning and supervised metric extraction with link constraints is to learn a distance such that the distances between data points in the same class are minimized and the distances between data points in different classes are maximized.

Recently, distance metric learning has been widely applied to many interesting machine learning problems (see [14, 15] for recent surveys), such as information retrieval, classification, computer vision and bioinformatics. Due to the importance of distance metric learning in metric-related pattern recognition tasks, a number of distance metric learning methods have been proposed [3]. These methods generally fall into two categories: methods based on eigen value optimization and methods based on convex or non-convex optimization. In the majority of these recent works on metric learning, one metric is constructed according to a feature-based description of a finite set of training objects, see survey [2]. In many of these works, the derived metric is the Euclidean distance between the projected feature vectors (sometimes imprecisely named Mahalanobis metric), and metric learning looks for the optimal projection. In this direction, the most famous methods are based on the ideas of Large-Margin nearest neighbor [3] and Information-theoretic metric learning [5]. The formulations of metric learning are similar to multiple kernel learning, which is intensively studied in the field of machine learning. The positive side of the approaches is that they directly try to increase one or another target value characterizing

the ‘compactness’ of classes. For example, they may require that an object’s nearest neighbor should be from the same class. The negative side is that the problems are often reduced to inconvenient optimization tasks which are difficult to study and solve. Accordingly, authors are often content with some number of iterations of gradient descent, without even checking convergence. In many cases, this is due to the incompatible constraints which make researchers to use soft-constraint approaches with many slack variables. In our approach, we can provably avoid using slack variables. In many works on metric learning, like [4], the derived distance may be calculated only to a training object, not between any pair of objects. Our approach combines distance functions and hence one can find the derived distance between among objects.

In our work, we have considered the size of the input data which is much higher than many of the existing machine learning methods. The distances characterize pairs of objects, and the variety of ways to compare objects to each other leads to large amounts of comparative information. Therefore, it is of some interest to formalize the metric aggregation problem in a way which reduces it to a convenient optimization problem. We propose one such formalization.

When researchers try to formalize the natural idea that intraclass distances should be small and interclass distances should be large, they usually introduce one scalar value characterizing the ratio of intraclass and interclass distances and solve an unconstrained optimization problem. Or, by analogy with SVM, both intraclass and interclass distances can be introduced into constraints and their greatest separation is required. We propose to consider intraclass and interclass distances asymmetrically, as in [3]: intraclass distances go to the objective function, interclass distances go to the constraints. That is why our approach is termed as Supervised Asymmetric Metric Extraction (SAME).

In [6,7] it was shown that it is convenient to theoretically investigate metric methods and effective to calculate when the derived metric is a linear combination of the original ones. Anyway, there exists no theory describing other convenient families of metric transformations, so in machine learning applications non-negative linear or convex combination of multiple metrics is ubiquitous.

3 Supervised Asymmetric Metric Extraction

We do not impose any assumptions on the object space except that it is being multimetric. Let N be the number of pseudo-metrics, and let ρ_1, \dots, ρ_N be the original pseudo-metrics on the object space. Let T be a finite sample of size M , and let x_1, \dots, x_M be the objects of the sample. We do not suppose that x_1, \dots, x_M have any particular structure other than that we can obtain all N distances between them.

Let the new metric r be a linear combination of the original ones: $r(x, y) = w_1\rho_1(x, y) + \dots + w_N\rho_N(x, y)$, where w_1, \dots, w_N are weights. It is guaranteed that r is a pseudo-metric if the weights w_1, \dots, w_N are non-negative [6]. In most domains, the scale of a distance has no specific semantics, so we do not demand the weights w_1, \dots, w_N meet convexity constraints. Very much the other way,

we will use the unity threshold for the derived metric no matter the scales of the original metrics.

Let S be the set of must-link pairs, let D be the set of cannot-link pairs. The pairs are unordered pairs of training object indices, and we do not consider reflexive pairs (i, i) . When the training objects have the actual labels y_1, \dots, y_M , we can set $S = \{(i, j) \mid y_i = y_j\}$ and $D = \{(i, j) \mid y_i \neq y_j\}$. We can say that S indicates intraclass pairs and D indicates interclass pairs.

The SAME problem is to minimize the average intraclass derived distance provided that all interclass derived distances are not less than 1 and the weights w_1, \dots, w_N are non-negative. The problem is formalized as (1).

$$\begin{aligned} & \sum_{(i,j) \in S} r(x_i, x_j) \rightarrow \min, \\ \text{s.t. } & r(x_i, x_j) \geq 1, \forall (i, j) \in D, \\ & w_n \geq 0, \forall n \in \{1, \dots, N\}. \end{aligned} \tag{1}$$

When we use the linear form of the derived metric r , (1) breaks down to (2).

$$\begin{aligned} & \sum_{(i,j) \in S} \sum_{n=1}^N w_n \rho_n(x_i, x_j) \rightarrow \min, \\ \text{s.t. } & \sum_{n=1}^N w_n \rho_n(x_i, x_j) \geq 1, \forall (i, j) \in D, \\ & w_n \geq 0, \forall n \in \{1, \dots, N\}. \end{aligned} \tag{2}$$

Changing the order of summation in the objective function, we get the conventional linear programming problem (3). We use the Iverson bracket notation here.

$$\begin{aligned} & \sum_{n=1}^N w_n \sum_{(i,j)} [y_i = y_j] \rho_n(x_i, x_j) \rightarrow \min, \\ \text{s.t. } & \sum_{n=1}^N w_n \rho_n(x_i, x_j) \geq 1, \text{ if } y_i \neq y_j, \\ & w_n \geq 0, \forall n. \end{aligned} \tag{3}$$

The formalization makes possible to consider the result as metric selection by positive weights. As the scales are ignored in the approach, so the weight rank has no semantics. The approach selects relevant metrics. To get rid of redundant metrics, it is advised to use unsupervised metric extraction in advance [1].

This is a linear programming problem, hence there is a wide range of methods and software to find the global optimum. The dual problem to (3) has the form (4).

$$\begin{aligned}
& \sum_{i,k:y_i \neq y_k} \lambda_{ik} \rightarrow \max, \\
\text{s.t.} \quad & \sum_{i,k:y_i \neq y_k} \lambda_{ik} \rho_n(x_i, x_k) \geq \sum_{i,j:y_i = y_j} \rho_n(x_i, x_j), \\
& \lambda_{ik} \geq 0, y_i \neq y_k.
\end{aligned} \tag{4}$$

Definition 1. *A pair of objects from D is conflicting, if for all the original metrics the distance between the objects is zero.*

Theorem 1. *In the linear programming problem, there is always an admissible solution, unless there are conflicting objects in the sample.*

It follows from the theorem that we can do without slack variables customary for SVM. A soft-margin extension is not required at all.

4 Biometric Applications

Let us consider two classification problems from computer vision domain. Both are related to biometric based person verification, i.e. in both cases the resulting classifier must provide which person (class) the given biometric information (object) corresponds to. Under biometric information we understand some numerical data, relatively small, obtained by a sensor or sensors and uniquely characterizing a person in which this data is obtained from. This data can be, for example, a person signature, as in the first problem, or an image of a person's palm, as in the second. In any case, it is quite natural to use distances (metrics or semimetrics) and similarity functions to compare these data objects and classify objects using their pairwise comparisons.

Further, in many cases, complexity of the data collected by the biometric sensors makes it difficult to select or construct the distance having the best class discrimination ability. In other cases, computation of such distance is unacceptably time demanding. Therefore, a researcher is often restricted to a set of simple, quickly calculated functions, capturing only a few discriminative features. Constructing a linear combination of them may be a good solution then. We will show that the proposed SAME method is able to find linear combinations of the original functions such that the resulting classification error is decreased comparing to the classification errors of the original functions.

4.1 Datasets and Original Metrics

For the signature verification problem we used a set of 1296 images of signatures, collected from 54 persons. Each person provided 24 images (slightly different, of course) of his/her signature. This sample was divided into train set, containing 810 images with 15 images taken from every of 54 persons, and test set, containing the remaining 486 images (9 from each person). Each image was labelled, i.e. it was known which person it had been taken from.

Each image of a signature was divided into eight columns of equal width. Histograms of each of these columns were calculated resulting in a feature vector containing eight groups of features. Features of the corresponding groups of two images were compared using the standard Earth Mover’s Distance (EMD). This produced a set of eight metrics, each comparing characteristics of an individual and small part of the whole image. As one can expect, none of these metrics can be good for classification separately, as each one “sees” only an eighth part of an image. At the same time, one can expect that some combination of these metrics can result in a considerably good discriminative function.

For the hand verification problem we used an extended dataset provided in [11]. It contains a total of 195 images of human palms taken from 35 persons. Taking the small sample size into account, we used LOO procedure to evaluate classification quality on this dataset rather than dividing it into train and test sets. Again, each image was labelled. For more details, see [11].

Hand palm images were compared against each other by calculating Hausdorff metric, measuring difference between skeleton curvature of the corresponding fingers (five values for a pair of images), by calculating difference between width of the corresponding fingers (five values). Hand palms were aligned before each comparison. As with signature verification, these metrics are not good enough for classification when used separately, but they are computationally efficient and are easily calculated even on an android platform.

4.2 Learned Metrics

In both cases we used the proposed SAME approach to obtain a new metric which was a linear combination of eight metrics in case of signature verification and ten metrics in case of hand verification. Only the 810 training samples were used to find the optimal linear coefficients in case of signature verification. In case of hand verification linear coefficients were calculated from the samples, remaining after excluding the test object in the LOO procedure. Hereafter, we will denote the original metrics, either for signature verification, or hand verification, as ρ_i , and the learned metric as r .

4.3 Performance Test

To compare the performance of original metrics with the learned ones, we used the simple 1NN method, because our goal is to evaluate the performance of metrics, rather than to build the best possible classifier. The error rate of the classification is assessed on the test samples, while all the training objects are used as prototypes in 1NN.

The table below shows the error rates for each of the original metrics and for the learned metric for signature verification problem.

One can see, that the metric, learned by the proposed SAME approach, does significantly better on the test sample, reducing the error rate almost fourfold, comparing to the original metrics. At the same time, the learning procedure

ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6	ρ_7	ρ_8	r
84.2%	89.7%	86.2%	89.5%	86.6%	84.8%	88.7%	85.0%	23.0%

implemented using Python is not computationally demanding, taking only 11.2 s to converge on Intel(R) Core(TM) i7-3520M CPU @ 2.90 GHz.

It must be noted, that the proposed SAME method is not designed to improve 1NN quality directly. And if the linear combination is trained specifically for 1NN, which can be done by a stochastic black-box optimization method, then the error rate can be reduced to 19.4%. But such brute force optimization can take 2 orders of magnitude longer.

For the hand verification problem the resulting improvement is even more dramatic. The following table shows the corresponding error rates.

ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6	ρ_7	ρ_8	ρ_9	ρ_{10}	r
21.0%	19.0%	17.4%	19.5%	31.8%	39.0%	31.3%	28.2%	40.0%	37.0%	1.5%

Again, the proposed SAME method constructs a metric that performs times better than the original metrics. And in this case, time required to calculate the linear coefficients is less than a second on the same machine.

Though it is possible to learn another linear combination, as for the signature verification, aiming to improve 1NN performance directly and get 0.5% error rate, it will take much more computational time.

5 Conclusion

In this paper we proposed a novel method, called Supervised Asymmetric Metric Extraction (SAME), for learning a linear combination of several original metrics, given their values on pairs of labeled train samples. We showed that this method possesses a number of attractive characteristics. Firstly, it relies only on the given values of the original metrics. In contrast to many metric learning approaches no features are required. Secondly, intraclass and interclass distances are treated asymmetrically with intraclass distances going to the objective function and interclass distances forming a system of constraints. This leads to the third characteristic. The resulting optimization problem is linear and can be effectively solved by a wide range of linear programming methods and software. Finally, it was proved that in the absence of conflicting objects the linear problem always has an admissible solution.

The proposed SAME method was evaluated on two biometric verification problems: signature verification and hand palm verification. Our experiments showed that metrics learned by SAME had significantly, several times lower error rates in comparison with the original, effectively calculated, but “weak” metrics.

It was also confirmed that the learning procedure is computationally effective, taking only seconds to calculate coefficients of the linear combination.

Theoretical and experimental results presented in this paper make the proposed method very promising and we are planning to compare it against other metric learning and metric combining methods, e.g. [12, 13], on various datasets in our future works.

Acknowledgments. This work was partially supported by Lomonosov Moscow State University research project “Algebraic, logical and statistical machine learning methods and their application in applied data analysis”, RFBR projects No 15-07-09214, 16-01-00196, 16-57-45054 and DST-RFBR Grant No. INT/RUS/RFBR/P-248.

References

1. Maysuradze, A.I., Suvorov, M.A.: Aggregation of multiple metric descriptions from distances between unlabeled objects. *J. Comput. Math. Math. Phys.* **57**(2), 350–361 (2017)
2. Bellet A., Habrard, A., Sebban, M.: A Survey on Metric Learning for Feature Vectors and Structured Data CoRR abs/1306.6709 (2013)
3. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.* **10**, 207–244 (2009)
4. Wang, Y., Lin, X., Zhang, Q.: Towards metric fusion on multi-view data: a cross-view based graph random walk approach. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management (CIKM 2013), pp. 805–810 (2013)
5. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: Proceedings of the 24th international conference on Machine learning, pp. 209–216. ACM (2007)
6. Maysuradze, A.I.: On optimal decompositions of finite metric configurations in pattern recognition problems. *J. Comput. Math. Math. Phys.* **44**(9), 1615–1624 (2004)
7. Maysuradze, A.I.: Homogeneous and rank bases in spaces of metric configurations. *J. Comput. Math. Math. Phys.* **46**(2), 330–344 (2006)
8. Mestetskiy, L.: Shape comparison of flexible objects: similarity of palm silhouettes. In: Proceedings of the 2nd International Conference on Computer Vision Theory and Applications (VISAPP), pp. 390–393 (2007)
9. Mestetskiy, L., Bakina, I., Kurakin, A.: Hand geometry analysis by continuous skeletons. In: Kamel, M., Campilho, A. (eds.) ICIAR 2011. LNCS, vol. 6754, pp. 130–139. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21596-4_14
10. Bakina, I., Mestetskiy, L.: Hand shape recognition from natural hand position. In: Proceedings of the IEEE International Conference on Hand-Based Biometrics. The Hong Kong Polytechnic University, Hong Kong, pp. 170–175 (2011)
11. Chernyshov, V., Mestetskiy, L.: Mobile computer vision system for hand-based identification. In: Pattern Recognition and Image Analysis, vol. 25, no. 2, pp. 209–214. Allen Press Inc., Cambridge (2015)
12. Suryanto, C.H., Hino, H., Fukui, K.: Combination of multiple distance measures for protein fold classification. In: 2nd IAPR Asian Conference on Pattern Recognition (ACPR), pp. 440–445. IEEE (2013)

13. Jang, D., Jang, S.-J., Lim, T.-B.: Distance combination for content identification system. In: 2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA), pp. 1–6 (2013)
14. Bellet, A., Habrard, A., Sebban, M.: Metric Learning, Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, San Rafael (2015)
15. Kulis, B.: Metric learning: a survey. In: Foundations and Trends in Machine Learning, vol. 5, no. 4, pp. 287–364 (2013)

Interval-Valued Writer-Dependent Global Features for Off-line Signature Verification

K. S. Manjunatha¹(✉), D. S. Guru², and H. Annapurna²

¹ Maharani's Science College for Women, Mysuru 570001, Karnataka, India
kowschik.manjunath@gmail.com

² Department of Studies in Computer Science, University of Mysore,
Manasagangothri, Mysuru 570006, Karnataka, India
dsg@compsci.uni-mysore.ac.in,
annapurnavmurthy@yahoo.co.in

Abstract. This work focuses on the proposal of a method for Off-line signature verification based on selecting writer-dependent global Features. 150 Global features of different categories namely geometrical, texture based, statistical and grid features for offline signatures are computed. Writer dependent features are selected through an application of a filter based feature selection method. Further, to preserve the intra-writer variations effectively, the selected features are represented by interval-valued data through aggregation of samples of each writer. Here in this work, we recommend creating two interval valued feature vectors for each writer. Decision on the test signature is accomplished by means of a symbolic classifier. In the first stage, we conducted experiments with writer dependent features by keeping a common dimension for all writers. Further, we conducted experiments with varying writer dependent feature dimension and threshold as done by a human expert. To demonstrate the effectiveness of the proposed approach extensive experimentation has been conducted on both CEDAR and MCYT offline signature datasets. The Error-rate obtained with the proposed model is low in comparison with many of contemporary models.

Keywords: Off-line signature · Global features · Interval valued data
Writer dependent features symbolic classifier

1 Introduction

During the last two decades, significant research has been carried out on authentication based on human body characteristics. Identity of a person can be established through physiological characteristics such as finger print, face, palm-print etc., or through behavioral characteristics such as signature, gait, voice etc. [1]. Signature verification can be carried out either offline or online [2]. Verification based on the features extracted from the geometry of the signature constitutes offline mode while the verification based on the dynamic properties such as velocity, pen-up and pen-down, pressure etc obtained from a capturing device constitutes online mode. Due to the non-availability of dynamic properties of writer, verification based on offline signature is complicated than that of online. Even today, authentication based on offline signature

finds applications in many of our day-to-day activities like transactions in bank, document attestation etc.

Many models which use different features, different classification techniques and different decision threshold have been proposed for offline signature verification during last few decades. Features used for offline signatures include geometric features [3–6], grid features [3], texture features [7], gradient features [8], statistical features [9], contour features [10] and centroid-based features [11]. During verification, the test signature is classified as either a genuine or a forgery based on the estimated similarity between the test signature and reference signatures of a claimed writer. Pattern recognition techniques such as support vector machine [8, 12–14] artificial neural network [4–6], hidden markov model [15, 16] and symbolic classifier [11] are adopted for verification.

The existing models share common characteristics that they consider a same set of discriminating features for all writers. But a human expert, while verifying a signature manually, considers different set of features for each individual. In addition, even the matching strategy adopted is also not same for all writers. This necessitates the usage of writer dependent features and writer- dependent classification technique for verification. Few works have been reported on the usage of writer-dependent classifier [17] and writer dependent threshold [18] for offline signature and limited attempts on the usage of writer dependent features in case of online signatures [19, 20]. Due to variations in the complexity of a signature, same number of features may not be necessary for all writers. Moreover, the threshold that decides the authenticity of a signature need not be same for all writers as some signatures are easy to forge when compared to others. Another challenging issue in case of off-line signature verification is to preserve the variations among the signature samples of a writer. In [11, 21], interval-valued symbolic representation has been well exploited for effectively preserving the intra-class variations and it has been well argued that interval-valued representation is more effective in preserving such variations.

With this backdrop, an approach has been recommended in this work for authentication based on off-line signatures, which exploits writer dependency at feature, feature dimension and threshold level and interval-valued symbolic representation for preserving intra-writer variations.

The paper is organized into 5 different sections. The various stages in the proposed method are discussed in Sect. 2. Experimental setup with the obtained results is presented in Sect. 3. We compare the performance of the proposed model with other similar models in Sect. 4. The overall conclusion is highlighted in Sect. 5.

2 Proposed Method

Following are the 5 distinct stages in the proposed method.

1. Feature computation
2. Selection of writer-dependent features
3. Fixation of feature dimension and decision threshold for individual writer
4. Clustering and creation of interval-valued feature vector
5. Verification

2.1 Feature Computation

Altogether, we have computed 150 global and grid features belonging to geometrical, statistical, texture and grid category as recommended in [3–5, 7, 9, 22, 23] after applying binarization, noise removal and thinning as pre-processing steps. The details of these 150 features computed are summarized in Table 1.

Table 1. List of 150 computed global features extracted and used

F#	Description	F#	Description	F#	Description
1	Homogeneity [7]	17	MaxHorihistgrm [22]	45	No. of connected components
2	Correlation [7]	18	Max.Vertical histogram [22]	46, 47, 48 and 49	Four area
3	Energy	19, 20, 21, 22, 23 and 24	Six fold surface features	50	Variance
4	Mean of a row vector [9]	25	Outline feature [4]	51	Normalized area of the signature [23]
5	Std. Dev. of row vector [9]	26	Fine ink distribution [4]	52	Core feature [4]
6	Mean of column vector [9]	27	Coarse ink distribution [4]	53	Slope
7	Std. Dev. of column vector [9]	28	High pressure_Region [4]	54	Ratio
8	Max. vertical projection	29, 30, 31, 32, 33, 34 35 and 36	Directional Frontiers [4]	55	Area [5]
9	Max. horizontal projection [3]	37	Total No_Of_Background_Pixels	56	Slope of the off-diagonal points of the bounding box
10	Hor.Proj . + Ver. Proj.	38	Height [3]	57 and 58	Global centroid
11	Slant angle [3]	39	Width [3]	59	Perimeter
12	Kurtosis [5]	40	Aspect ratio [23]	60, 61, 62 and 63	Center of mass
13	Skewness [5]	41	Orientation	64	Major axis length
14	Baseline shift [3]	42	Wrinkleless	65, 66 and 67	Tri surface features
15	No. of end points [22]	43	No. of branch points	68	Minor axis length
				69	Center feature
16	Euler_Number	44	Mean	70 to 150	Grid features

2.2 Selection of Writer-Dependent Features

In order to select features which are relevant for each writer, we have adopted the feature selection approach proposed in [24] which is filter-based, computationally less expensive and suitable for especially multi-cluster data. The motive to adopt this method is its suitability to multi-clustered data like signatures where the signatures of a writer form some natural cluster. Here, for every feature, its relevancy is decided based on computing a relevancy score. The score depicts the capability of the individual feature in maintaining the structure of cluster of the respective writer.

Let n denote the number of training signatures contributed by the writer W_i ($i = 1, 2, \dots, N$) where N denotes the number of writers. For each signature, P global features are computed which results in a data matrix of dimension $n \times P$ for a writer. For each of the P feature, a relevancy score is computed through the application of a multi-cluster feature selection method (MCFS) [24]. The computed scores are sorted in the decreasing order and we recommend selecting only the d features with highest score. The selected d features vary from a writer to a writer. We preserve the indices of all d features selected for every writer in the knowledgebase which is needed during verification. For more details on MCFS method, reader can refer [24].

2.3 Fixation of Feature Dimension and Threshold for Individual Writer

The two parameters namely feature dimension and the threshold to be used for each writer are determined empirically as follows. After computing the relevancy of all features for each writer as discussed in Sect. 2.2, we estimate the FAR and FRR for different values of d under varying similarity threshold from 0.1 to 1.0 and the EER is computed. The decision on the feature dimension and the threshold for individual writer is based on a minimum EER criterion. That is, the corresponding value of d and the threshold at which the obtained EER is the lowest is fixed as the suitable feature dimension and decision threshold for that writer. We arrive at the decision on these two parameters based on the result of 10 trials with random selection of training signatures in each trial. Table 2 shows the feature dimension and the threshold for sample 5 writers of MCYT dataset (With 30% of genuine signatures for training purpose).

Table 2. Feature dimension and the threshold obtained for sample 5 writers of MCYT dataset along with EER obtained.

Writer-ID	Feature dimension (d)	Threshold	EER
04	45	0.5	12.73
33	05	0.3	05.71
45	25	0.5	12.00
61	35	0.5	04.00
72	05	0.5	09.09

2.4 Clustering and Creation of Interval-Valued Feature Vector

After selecting the writer-dependent features as discussed in Sect. 2.3, we recommend clustering the training signatures of each writer based on the selected features. For the purpose of clustering, we have adopted Fuzzy-C means method, as it is easy to implement and also distribution free. The reason for clustering the signatures of a writer is to have multiple representatives for a writer and to assimilate the variations in the feature value of signatures of the same writer. For each cluster, a single representative is created through symbolic representation. In this work, we have adopted the interval valued symbolic representation proposed in the work [11]. The interval-valued symbolic representation for a writer is computed as follows

Let $\{Sig_1, Sig_2, Sig_3, \dots, Sig_n\}$ represent the n signatures belonging to a cluster X_{cl} , ($cl = 1, 2, \dots, M_c$) where M_c is the number of clusters for each writer.

Let $\{f_{i1}, f_{i2}, \dots, f_{id}\}$ be the feature vector characterizing signature sample Sig_i of the cluster X_{cl} ($cl = 1, 2, \dots, M_c$). Let $Mean_{clK}$ ($K = 1, 2, \dots, d$) denote the mean and Std_{clK} ($K = 1, 2, \dots, d$) denote the standard deviation of the K^{th} feature value considering all the n training signatures of the cluster X_{cl} .

We recommend representing the K^{th} feature of a cl^{th} cluster as an interval-valued feature as $[f_{clK}^-, f_{clK}^+]$, Where $f_{clK}^- = Mean_{clK} - Std_{clK}$ and $f_{clK}^+ = Mean_{clK} + Std_{clK}$. The interval $[f_{clK}^-, f_{clK}^+]$ denotes respectively the lower limit and upper limits of a K^{th} feature value of a cl^{th} cluster in the knowledgebase.

Similarly, all the d features are represented in the form of an interval-valued thereby resulting in a reference feature vector for a cluster X_{cl} as $RF_{cl} = \{[f_{cl1}^-, f_{cl1}^+], [f_{cl2}^-, f_{cl2}^+], \dots, [f_{cld}^-, f_{cld}^+]\}$. Altogether in the knowledgebase we need to store only NXM_c .

2.5 Verification

In order to decide whether a test signature claimed to be of a writer W_i is genuine or forgery, the test signature is described by means of a set of P features of crisp type as $F_q = \{f_{q1}, f_{q2}, \dots, f_{qP}\}$. Out of the P features, we consider only d features for comparison purpose by retrieving the indices of all the d features from the knowledgebase. A count of the number of features of a test signature that falls within the corresponding interval of a reference signature is estimated and denoted as the acceptance count (A_c) which is computed as follows

$$A_c = \sum_{K=1}^d C(f_{qK}, [f_{clK}^-, f_{clK}^+]) \text{ Where}$$

$$C(f_{qK}, [f_{clK}^-, f_{clK}^+]) = \begin{cases} 1 & \text{if } (f_{qK} > = f_{clK}^- \text{ and } f_{qK} < = f_{clK}^+) \\ 0 & \text{otherwise} \end{cases} \text{ for any}$$

$$cl = 1, 2, \dots, M_c$$

3 Experimentation and Results

All experimentation are conducted on both CEDAR [25] and MYCT [26] off-line signature datasets. CEDAR dataset consists of 24 genuine and 24 skilled forgeries from 55 writers. The MCYT dataset contains 15 genuine and 15 skilled forgeries from 75 writers.

In the first stage, we conducted the experiment with a common feature dimension and threshold for all writers under varying percentage of genuine training signatures from 30% to 70%. The genuine signatures not used for training and all the skilled forgeries are used for testing purpose. For random forgery, one genuine signature of every other writer is considered as random forgery. We conducted 10 different trials with random selection of training and testing signatures. The Equal Error Rate (EER) under varying percentage of training signatures based on a common feature dimension and threshold are shown in Tables 3 and 4 respectively for CEDAR and MCYT datasets.

Table 3. Error rate obtained on CEDAR dataset (With common feature dimension and threshold)

Number of training samples	Skilled_Forgery	Random_Forgery
07	24.64	12.73
10	23.54	12.28
12	21.22	10.66
14	18.26	10.43
17	20.44	10.88

Table 4. Error rate obtained on MCYT dataset (With common feature dimension and threshold)

Number of training samples	Skilled_Forgery	Random_Forgery
05	21.06	11.54
06	19.33	08.79
08	17.20	08.19
09	14.73	07.99
10	14.64	06.32

Further, we conducted verification based on the usage of feature dimension and threshold which are fixed for each writer as discussed in Sect. 2.3 and the EER obtained are shown in Tables 5 and 6 corresponding to CEDAR and MCYT datasets respectively.

Table 5. Error rate obtained on CEDAR dataset (With variable feature dimension and threshold)

Number of training samples	Skilled_Forgery	Random_Forgery
07	14.82	08.28
10	16.39	06.73
12	14.87	06.28
14	07.66	05.41
17	13.04	05.61

Table 6. Error rate obtained by the proposed model on MCYT dataset (With variable feature dimension and threshold)

Number of training samples	Skilled_Forgery	Random_Forgery
05	13.67	06.97
06	13.17	05.61
08	11.24	04.24
09	09.96	03.86
10	09.53	03.24

4 Comparative Analysis

In this work, the performance of the proposed model is compared with other contemporary models for off-line signature verification. We have used EER as a measure to compare the performance of our model with other models. As the number of training samples used also varies and hence comparison of different models is difficult. For comparative analysis, we considered the models evaluated on CEDAR and MCYT datasets and the results are tabulated in Tables 7 and 8 respectively. In Tables 7 and 8, the error rate obtained with only skilled forgeries are reported as most of the existing models have reported the results for skilled forgeries only and EER with random forgeries is lower than that of skilled forgeries as the former one is easy to detect.

Table 7. EER of the various offline signature models on CEDAR dataset

Model	Number of training samples	EER
[18]	24	21.90
[29]	16	07.90
[27]	24	08.33
[30]	12	07.84
[14]	12	05.60
Proposed	14	07.66

Table 8. EER of the various off-line signature models on MCYT-75 dataset

Model	Number of training samples	EER
[35]	05	22.4
[35]	10	20.00
[33]	05	15.02
[7]	05	11.28
[7]	10	07.23
[34]	05	12.02
[10]	05	10.18
[10]	10	06.44
[32]	05	13.86
[32]	10	09.87
[31]	05	13.44
[31]	10	09.86
[28]	05	03.58
[28]	10	02.87
Proposed model (Common feature dimension and threshold)	05	13.67
Proposed model (Variable feature dimension and threshold)	10	09.53

In Table 7, the best result reported is considered for comparative study. In Table 7, it can be observed that the error rate obtained with the proposed model is lower than the error rate of the existing models except [14, 28]. Ours is first of its kind on the usage of writer dependent features for offline signature verification while other models are based on the features which are writer-independent. This work is an initial attempt on exploitation of writer dependency for verification purpose. Further, exploration in this direction only opens new avenues but further minimizing the error rate can be investigated.

Generally, when the availability of training samples are more, a writer-dependent model results in better performance when compared to a writer-independent model. This is due to the fact, that a writer-dependent model needs more training samples as the characteristics vary from a writer to a writer. Hence, from Table 8, it can be observed that error rate obtained with the proposed model is lower than most of the existing models except [7, 10, 28] with 10 training signatures. The performance of model with 5 training signatures is poor when compared to other models. But there are number of applications such as banks and credit card transaction where it is not hard to collect the signatures as the customer provides his/her signature during every transaction.

5 Conclusion

A method for off-line signature verification is proposed based on the selection of features which may vary from an individual to an individual. Relevant features for each writer are selected through the application of an efficient feature selection method. Further, the notion of interval-valued symbolic representation has been adopted for representing the features which are selected for each writer. In order to have a compact representation, the samples of each writer are aggregated using fuzzy-C means clustering method and interval-valued data representation. In addition, writer dependency at dimension and threshold level also have been exploited which resulted in further reduction of error rate. The effectiveness of the model is demonstrated with rigorous experimentation on two standard benchmarking datasets. The error rate obtained shows the effectiveness of the method proposed for verification purpose.

References

1. Plamondon, R., Lorette, G.: Automatic signature verification and writer identification - the state of the art. *Pattern Recogn.* **2**, 107–131 (1989)
2. Jain, A.K., Griess, F.D., Connell, S.D.: On-line signature verification. *Pattern Recogn.* **35**, 2963–2972 (2002)
3. Qi, Y., Hunt, B.R.: Signature verification using global and grid features. *Pattern Recogn.* **27**(12), 1621–1629 (1994)
4. Huang, K., Yan, H.: Off-line signature verification based on geometric feature extraction and neural network classification. *Pattern Recogn.* **30**, 9–17 (1997)
5. Karouni, A., Daya, B., Bahlak, S.: Offline signature recognition using neural networks approach. *Procedia Comput. Sci.* **3**, 155–161 (2011)
6. Hatkar, P.V., Salokhe, B.T., Malgave, A.A.: Off-line handwritten signature verification using neural network. *Int. J. Innov. Eng. Res. Technol.* **2**(1), 1–5 (2015)
7. Vargas, J.F., Ferrer, M.A., Travieso, C.M., Alonso, J.B.: Off-line signature verification based on grey level information using texture features. *Pattern Recogn.* **44**, 375–385 (2011)
8. Nguyen, V., Kawazoey, Y., Wakabayashiy, T., Pal, U., Blumenstein, M.: Performance analysis of the gradient feature and the modified direction feature for off-line signature verification. In: *Proceeding of IEEE 12th International Conference on Frontiers in Handwriting Recognition*, pp. 303–307 (2010)
9. Mhatre, P.M., Maniroja, M.: Offline signature verification based on statistical features. Published in *Proceedings of International Conference & Workshop on Emerging Trends in Technology*, pp. 59–62 (2011)
10. Gilperez, A., Fernandez, F.A., Pecharroman, S., Fierrez, J., Garcia, J.O.: Off-line signature verification using contour features. In: *ICFHR*, pp. 1–6 (2013)
11. Prakash, H.N., Guru, D.S.: Relative orientations of geometric centroids for off-line signature verification. In: *ICAPR*, pp. 201–204 (2009)
12. Lv, H., Wang, W., Wang, C., Zhuo, Q.: Off-line Chinese signature verification based on support vector machines. *Pattern Recongn. Lett.* **26**, 2390–2399 (2005)
13. Parodi, M., Gomez, J.C., Belaid, A.: A circular grid-based rotation invariant feature extraction approach for off-line signature verification. In: *ICDAR*, pp. 1289–1293 (2011)

14. Guerbai, Y., Chibani, Y., Hadjadji, B.: The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters. *Pattern Recogn.* **48**, 103–113 (2015)
15. Coetzer, J., Herbst, B.M., duPreez, J.A.: Offline signature verification using the discrete radon transform and a hidden Markov model. *EURASIP J. Appl. Sig. Process.* **4**, 559–571 (2004)
16. Daramola, D.S.A., Ibiyemi, P.T.S.: Offline signature recognition using hidden markov model (HMM). *Int. J. Comput. Appl.* **10**, 17–22 (2010)
17. Eskander, G.S., Sabourin, R., Granger, E.: Hybrid writer-independent –writer –dependent offline signature verification system. *IET Biometrics* **2**(4), 169–181 (2013)
18. Srihari, S.N., Xu, A., Kalera, M.K.: Learning strategies and classification methods for off-line signature verification. In: *IWFHR*, pp. 1–6 (2004)
19. Guru, D.S., Manjunatha, K.S., Manjunath, S.: User dependent features in online signature verification. In: Swamy, P., Guru, D. (eds.) *Multimedia Processing, Communication and Computing Applications. LNEE*, vol. 213, pp. 229–239. Springer, New Delhi (2013). https://doi.org/10.1007/978-81-322-1143-3_19
20. Manjunatha, K.S., Manjunath, S., Guru, D.S., Somashekara, M.T.: Online signature verification based on writer dependent features and classifiers. *Pattern Recogn. Lett.* **80**, 129–136 (2016)
21. Alaei, A., Pal, S., Pal, U.: An efficient signature verification method based on interval symbolic representation and Fuzzy similarity measure. *IEEE Trans. Inf. Forensics Secur.* **12**(10), 2360–2372 (2017)
22. Ramachandra, A.C., Rao, J.S., Raja, K.B., Venugopla, K.R., Patnaik, L.M.: Robust offline signature verification based on global features. Published in *IEEE International Advance Computing Conference*, pp. 1173–1178 (2009)
23. Kruthi, C., Shet, D.C.: Offline signature verification using support vector machine. In: *IEEE Transactions* (2014). <https://doi.org/10.1109/ICVIP.2014.5>
24. Cai, D., Zhang, C., He, X.: Unsupervised feature selection for multi-cluster data. In: *International Conference on Knowledge Discovery and Data Mining*, pp. 333–342 (2010)
25. Kalera, M.K., Srihari, S., Xu, A.: Offline signature verification and identification using distance statistics. *Int. J. Pattern Recogn. Artif. Intell. (IJPRAI)* **18**(7), 1339–1360 (2004)
26. Garcia, O.J., Aguilier, J.F., Simon, D.: MCYT baseline corpus: a bimodal database. In: *IEE Proceedings Vision, Image and Signal Processing*, pp. 395–401 (2003)
27. Kumar, R., Sharma, J.D., Chanda, B.: Writer-independent off-line signature verification using surroundedness feature. *Pattern Recogn. Lett.* **33**, 301–308 (2012)
28. Hafemann, L.G., Sabourin, R., Oliveira, L.S.: Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recogn.* **70**, 163–176 (2017)
29. Chen, S., Srihari, S.: A new off-line signature verification method based on graph. In: *Proceedings of 18th International Conference on Pattern Recognition*, pp. 869–872 (2006)
30. Bharathi, R., Shekar, B.: Off-line signature verification based on chain code histogram and support vector machine. Published in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2063–2068. <https://doi.org/10.1109/ICACCI.2013.6637499>
31. Soleimani, A., Araabi, B.N., Fouladi, K.: Deep multitask metric learning for offline signature verification. *Pattern Recogn. Lett.* **80**, 84–90 (2016)
32. Ooi, S.Y., Teoh, A.B.J., Pang, Y.H., Hiew, B.Y.: Image-based handwritten signature verification using hybrid methods of discrete Radon transform, principal component analysis and probabilistic neural network. *Appl. Soft Comput.* **40**, 274–282 (2016)

33. Wen, J., Fang, B., Tang, Y.Y., Zhang, T.: Model-based signature verification with rotation invariant features. *Pattern Recogn.* **42**, 1458–1466 (2009)
34. Ferrer, M.A., Vargas, J.F., Morales, A., Ordóñez, A.: Robustness of offline signature verification based on gray level features. *IEEE Trans. Inf. Forensic Secur.* **7**(3), 966–977 (2012)
35. Alonso-Fernandez, F., Fairhurst, M.C., Fierrez, J., Ortega-García, J.: Automatic measures for predicting performance in off-line signature. In: *ICIP*, vol. I, pp. 369–372 (2007)

Despeckling with Structure Preservation in Clinical Ultrasound Images Using Historical Edge Information Weighted Regularizer

Rahul Roy¹, Susmita Ghosh², Sung-Bae Cho³, and Ashish Ghosh¹(✉)

¹ Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700108, India
{rahulroy_r, ash}@isical.ac.in

² Department of Computer Science and Engineering,
Jadavpur University, Kolkata 700032, India
susmitaghoshju@gmail.com

³ Soft Computing Laboratory, Department of Computer Science,
Yonsei University, Seoul 120749, Korea
sbcho@cs.yonsei.ac.kr

Abstract. This article presents a de-speckling technique for clinical ultrasound images with an aim to preserve the fine structural information and region boundaries in images. The algorithm generates restored images by minimizing the variational energy on them. To compute variational energy, a weighted total variation based method is proposed where the weights are determined from both historical (previous/earlier time stamp) as well as instantaneous oriented structural information of images. This helps in defining the anisotropy at edges in the image which, in turn, helps in identifying homogenous regions on it. Moreover, the method is able to preserve the vague echo-textural differences which might be of clinical importance but may get destroyed due to smoothing operations. To elicit effectiveness, comparative analysis of the proposed approaches have been done with four state-of-the-art techniques on both *in silico* and *in vivo* ultrasound images using four standard measures (two for phantom images and two for clinical ultrasound images). Qualitative and quantitative analysis reveals the promising performance of the proposed technique.

Keywords: Clinical ultrasound image · Speckle de-noising
Bayesian MAP · Bregman alternate method of multipliers

1 Introduction

Speckles are inherent noise in ultrasound images which arise due to interference of scattered signals from the interface of tissues with the reflected ones [17]. They tend to obscure the structural details and hence induce heterogeneity in images. Thus designing automated tool for localization of organ becomes a challenging task. As a result, despeckling of images is considered to be an integrated pre-processing module for such automated tool [14, 17].

Several researchers have been working on despeckling of clinical ultrasound images [3,23]. Many filtering [3,23] and Partial differential equation (PDE) [12,25] based approaches have been developed in this regard. However, most of these algorithms fail to balance smoothing and preservation of edges in images. Moreover, they suffer from broadening of edges as they evolve over time. Recently, variational energy minimization based de-speckling strategy has been in the spotlight. This approach is characterized by a regularization term (measuring the variational vitality) and an information fitting term (measuring the residual vitality) After a pioneering work from Aubert and Aujol [1], many researchers [9,19] have been focusing on defining suitable data fitting term so that the optimization function is a convex one. It is to be noted that the said methodologies consider total variation (TV) as a regularizer, which yields piecewise smooth images. Though TV yields sharp edges, the restored images experience ill effects of staircase artifacts, expansion of edges and the methods do not save the directionality of edges in images. To overcome this effect, many non convex [8] and total high order variation [7] and their combination [11,22] based regularizer have been developed. The TV as regularizer do not takes the echo texture semantics into account due to which it is difficult to identify the false variation due to speckle and vague structures. Furthermore, as the TV are computed on instantaneous edge variation, there is a probability for failure in preserving the vague structures.

This motivated use to define a regularizer that could take into account the above information for preserving vague edges along with their directionality. This direction specific information of tissue textures and edges can be of importance for diagnosis and tissue characterizations. In this article, a historical and instantaneous local oriented structure information based weighing mechanism is proposed. This oriented structure information introduces anisotropy in TV regularizer and historical edge information controls the loss of vague tissue boundary across the scale space. A block diagram of the proposed approach is given in Fig. 1.

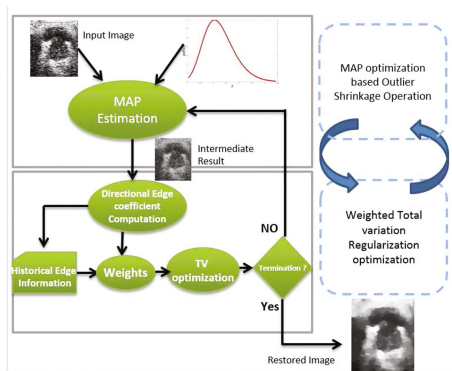


Fig. 1. Block diagram of proposed method

The proposed algorithm is compared with four state-of-the-art techniques (namely, oriented speckle reducing anisotropic diffusion filters (OSRAD) [12], optimized Bayesian non-local mean (OBNLM) [3], anisotropic diffusion filter with memory based on speckle statistics (ADMMS) [16], Rayleigh based total variation (RayleighTV) [9]) in different *in silico* and clinical ultrasound images. Two performance measures viz. figure of merit (FoM) and structural similarity index (SSIM) are used for comparing the results on *in silico* images and two measures namely, speckle suppression index (SSI) and effective number of look-ups (ENL) are used for comparing the result of clinical ultrasound images. To demonstrate the effectiveness of the proposed algorithm for de-noising images, results are analyzed both qualitatively and quantitatively. Experiment reveals the promising performance of the proposed algorithm in reducing speckles while preserving structural details in images.

The rest of the article is organized as follows. In Sect. 2, proposed de-speckling procedure is described. Experimental results and analysis are provided in Sect. 3. Finally, conclusion and future scope are put in Sect. 4.

2 Proposed Approach

In this approach, the de-speckling strategy follows a Maximum A Posteriori (MAP) estimation based framework for recovering piecewise smooth image (original signal) from a given noisy image. The direction and vague structures are preserved using present and historical (earlier time stamp) local oriented structural information from images. We term the algorithm as “despeckling with historical structure information weighted total variation” (DsHSIWTV).

Let $Z \in \mathbb{R}_+^{m \times n}$ be the noisy image and A be its noise free reflectivity which is of the same size as Z . Then the multiplicative noise model can be written as [2, 24]

$$Z(\mathbf{x}) = A(\mathbf{x})N(\mathbf{x}) \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^d$ is the spatial location in d dimensional space and N represents noise and is of the same size as Z .

It may be noted that, clinical ultrasound images undergo a non-linear transformation to fit the dynamic range of display. Thus the noise model defined in Eq. (1) transforms to (as done in [5])

$$Y(\mathbf{x}) = D \log(Z(\mathbf{x})) + G. \quad (2)$$

Here Y is the log compressed image which is of the same size as Z . D and G are parameters representing the dynamic range of input and linear gain of the amplifier, respectively. Substituting Eq. (1) in (2) and rearranging we get,

$$Y(\mathbf{x}) = U(\mathbf{x}) + W(\mathbf{x}), \quad (3)$$

where $U(\mathbf{x}) = \log [A(\mathbf{x})]^D + G$ and $W(\mathbf{x}) = \log [N(\mathbf{x})]^D$.

Here we assume that the log compressed speckle noise W follows Generalized Fisher Tippett distribution given as

$$p_W(w) = \frac{2se^m}{\sqrt{\frac{2\pi}{m}}} e^{m[(2sw - \ln \Omega) - e^{(2sw - \ln \Omega)}]}, \quad s = \frac{1}{D}; \quad (4)$$

where m and s are shape parameters and Ω is the scale parameter. Thus this distribution is used to define the posterior probability of the MAP.

In Bayesian framework, let U be a random variable representing the piecewise smooth reflective image that can be estimated from the observed image Y . Let, u and y respectively, be the realizations of the random variable U and Y . η is realization of noisy random variable N . Bayes' rule in logarithmic form can be written as

$$-\log(p_{(U|Y)}(u(\mathbf{x})|y(\mathbf{x}))) = -\log p_N(\eta(\mathbf{x})) - \log p(u(\mathbf{x})). \quad (5)$$

The likelihood in Eq. (5) is computed considering the distribution defined in Eq. (4). So, substituting the instance of noise $\eta(\mathbf{x})$ with $(y(\mathbf{x}) - u(\mathbf{x}))$ in (5), we get

$$\log(p_W(y(\mathbf{x}) - u(\mathbf{x}))) \propto \sum_{\mathbf{x}} m(2su(\mathbf{x}) + \frac{1}{2\sigma^2} e^{2s(y(\mathbf{x}) - u(\mathbf{x}))}). \quad (6)$$

The prior probability $p(u)$ can be modeled with Markov Random Field which explains the statistical properties of the logarithm of a true image [13]. Thus,

$$-\log(p_U(u)) \propto \sum_{\mathbf{x}} \lambda(\mathbf{A}(\mathbf{x}) (|\nabla u(\mathbf{x})|)), \quad (7)$$

where weight \mathbf{A} defines the anisotropy at each pixel position of the image.

Putting Eqs. (6) and (7) in the MAP estimation criterion, defined in Eq. (5), unconstrained optimization function can be written as

$$\underset{\hat{u}}{\operatorname{argmin}} f(u) = \sum_{\mathbf{x}} m \left(2su(\mathbf{x}) + \frac{1}{2\sigma^2} e^{2s(y(\mathbf{x}) - u(\mathbf{x}))} \right) + \lambda \sum_{\mathbf{x}} \mathbf{A}(\mathbf{x}) (|\nabla u(\mathbf{x})|), \quad (8)$$

where the regularizer λ is determined empirically.

To incorporate local oriented and historical structural information, \mathbf{A} is modeled with delay differential equation (DDE) given as [16]

$$\frac{d\mathbf{A}}{dt} = \frac{1}{\tau(U)} [\mathbf{A}((\mathbf{x}), t) - S\{\mathbf{D}((\mathbf{x}), t)\}], \quad (9)$$

where $\tau((\mathbf{x}))$ is the relaxation time and \mathbf{D} is the instantaneous diffusion tensor which is computed as [12]

$$\mathbf{D}_t((\mathbf{x})) = \begin{pmatrix} C(q, t) & 0 \\ 0 & C_{max} \end{pmatrix}.$$

Here, $C : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ is a continuous function which takes large values at the edges in order to cease smoothing near the edges. Mathematically,

$$C(q, t) = e^{\left\{ -\frac{[q^2((\mathbf{x}), t) - q_0^2(t)]}{[q_0^2(t)(1 + q_0^2(t))]} \right\}} \quad (10)$$

where q represents the instantaneous coefficient of variation [25]

$$q = \sqrt{\frac{0.5(|\nabla u(\mathbf{x})|/u(\mathbf{x}))^2 + 0.0625(\nabla^2 u(\mathbf{x})/u(\mathbf{x}))^2}{[1 + 0.25(\nabla^2 u(\mathbf{x})/u(\mathbf{x}))]^2}} \quad (11)$$

and

$$q_0(t) = \frac{\sqrt{\text{var}(u((\mathbf{x}), t))}}{u((\mathbf{x}), t)}.$$

C_{max} defines the amount of smoothing that is carried out along the boundary. It is computed as

$$C_{\max}(\mathbf{x}) = L_{xx}(\mathbf{x}) + L_{yy}(\mathbf{x}) + \sqrt{(L_{xx}(\mathbf{x}) - L_{yy}(\mathbf{x}))^2 + 4L_{xy}^2(\mathbf{x})} \quad ; \quad (12)$$

The diffusion tensor $S\{\mathbf{D}((\mathbf{x}), t)\}$ is defined as

$$S\{\mathbf{D}((\mathbf{x}), t)\} = p_c((\mathbf{x}), t)\mathbf{D}((\mathbf{x}), t).$$

Here $p_c((\mathbf{x}), t)$ denotes the degree of belonging of a pixel to the echotexture of a tissue. As it is known from literature in medicine, an ultrasound echotexture can be grouped into three categories (namely, echogenic, hypo-echogenic and hyper-echogenic). In this context, a fuzzy C-means (FCM) algorithm is used to cluster the intensity of the pixel into three echotextures and a cluster map is constructed in decreasing order of intensity of cluster centers. The final echotexture map is generated by assigning each pixel \mathbf{x} the intensity of the cluster center to which it belongs.

The DDE of Eq. (9) can be easily integrated to find the solution as

$$\mathbf{A}((\mathbf{x}), t) = S\{\mathbf{D}(\mathbf{x}, 0)\}e^{-\frac{t}{\tau}} + \int_0^t e^{-\frac{s-t}{\tau}} S\{\mathbf{D}((\mathbf{x}), s)\}ds, \quad (13)$$

where τ is the relaxation time which is responsible for selective historical edge preservation mechanism. It specifies the contribution of historical edge information in defining the weights at each location \mathbf{x} . Thus τ is defined as

$$\tau(p_c((\mathbf{x}), t)) = \frac{1 - p_c((\mathbf{x}), t)}{[p_c((\mathbf{x}), t)]^n}. \quad (14)$$

Equation (13) is continuous in nature and needs to be discretized. Hence we adopt discrete form for Eq. (9) and is given as

$$\mathbf{A}^{t+1}(\mathbf{x}) = \frac{1}{1 + \beta^t} [\beta^t \mathbf{A}^t(\mathbf{x}) + S\{\mathbf{D}^t(\mathbf{x})\}], \quad (15)$$

where $\beta^t = \frac{(p_c(\mathbf{x}), n\delta t)}{\delta t}$ and δt is the time interval between two time instances. \mathbf{A}^{t+1} denotes the updated weights at time $t + 1$.

The functional $f(u)$ is minimized by decomposing into two sub-problems using Bregman alternate method of multipliers [21]. Such decomposition results into following two functional forms

$$\underset{\hat{u}}{\operatorname{argmin}} f(u) = \sum_{\mathbf{x}} \left(m(2su(\mathbf{x}) + \frac{1}{2} e^{2s(y(\mathbf{x}) - u(\mathbf{x}))}) + \frac{\mu}{2} (u(\mathbf{x}) - v^t(\mathbf{x}) - d^t(\mathbf{x}))^2 \right), \quad (16)$$

$$\underset{\hat{v}}{\operatorname{argmin}} f(v) = \sum_{\mathbf{x}} \left(\lambda \mathbf{A}(\mathbf{x}) (|\nabla v(\mathbf{x})|) + \frac{\mu}{2} (u^t(\mathbf{x}) - v(\mathbf{x}) - d^t(\mathbf{x}))^2 \right), \quad (17)$$

$$\text{with } d^{t+1} = d^t(\mathbf{x}) + (U^{t+1}(\mathbf{x}) - V^{t+1}(\mathbf{x})); \quad (18)$$

A closed form solution, derived for the first functional, is written as

$$\hat{u}(\mathbf{x}) = \frac{\nu}{0.25} (u(\mathbf{x})) + (1 - \frac{\nu}{0.25}) (v(\mathbf{x}) + d(\mathbf{x})) - \nu \left(2sm \left(1 - \frac{1}{2\sigma^2} e^{2s(f(\mathbf{x}) - u(\mathbf{x}))} \right) \right). \quad (19)$$

The second function can be solved using the generalized Chambolle projection algorithm [6]. These two functions are solved iteratively to obtain the final restored images. Here the number of iterations (pre-specified) is used as stopping criteria for the algorithm.

3 Experimental Analysis

Experiments were carried out on an Intel Core i7 3.40 GHz processor, with 16 GB RAM, Windows operating system and Matlab programming environment. Though experiments are conducted on various phantom and clinical US images, quantitative results are reported on a test suite of 5 images with various scattering scenario. However, due to the space limitation visual results of only two images are discussed here.

As mentioned, performance of the proposed approach is compared with those of different popular de-speckling methodologies present in the literature. The algorithms, taken into consideration, include filtering approaches like optimized Bayesian non-local mean filters (OBNLM) [3]; Partial differential equation (PDE) based approaches like oriented speckle reducing anisotropic diffusion (OSRAD) [12] & anisotropic diffusion filter with memory model based on speckle statistic (ADMMS) [16] are also considered. Along with it, total variation with Rayleigh distribution (RayleighTV) [9] is used for evaluation.

Quantitative analysis of results obtained using different de-speckling methods is made based on the measures proposed in [14]. Here the structural features are assessed using structural similarity index (SSIM) and Pratt's figure of merit (FOM) [4]. Higher value of the indices indicate better performance of the algorithm.

Let g represent the original image and \hat{u} represent the de-noised one. The measures are as follows:

$$SSIM = \frac{(2\bar{g}\bar{u} + c_1)(2\sigma_{g\hat{u}} + c_2)}{(\bar{g}^2 + \bar{u}^2 + c_1)(\sigma_g^2 + \sigma_{\hat{u}}^2 + c_2)}, \quad (20)$$

where $c_1 = 0.001dr$ and $c_2 = 0.003dr$ with $dr = 255$ representing the dynamic range of US images.

$$FOM = \frac{1}{\max(N_g, N_u)} \sum_{i=1}^{N_g} \frac{1}{1 + d_i^2 \alpha}, \quad (21)$$

where N_g and N_u are the number of edge pixels in ground truth and de-noised images respectively. d_i is the distance between the i^{th} pixel in the ground truth image and the neighboring pixel of the denoised image. Here α represent a constant used to penalise displaced edges.

There are a few direct measures for quantitative analysis of de-noising algorithms on clinical US images. These include speckle suppression index (SSI) [4] which is defined as the ratio of variance of homogeneous region (as specified by user) before and after de-noising. The lower value indicates better suppression of speckle noise. It is defined as

$$SSI = \frac{\sqrt{\text{var}(\hat{u})}}{\text{mean}(\hat{u})} \times \frac{\text{mean}(y)}{\sqrt{\text{var}(y)}}. \quad (22)$$

The effective number of lookup (ENL) [4] defined as the ratio of mean and standard deviation of filtered image, is also considered in the present investigation, where

$$ENL = \frac{\text{mean}(\hat{u})}{\sqrt{\text{var}(\hat{u})}}. \quad (23)$$

For experimentation, 10000 random regions were selected and the SSI and ENL were computed to avoid bias in evaluating the performance of the algorithms. The experiment is repeated 100 times and the average result is reported in this article.

3.1 Dataset

Facsimile US images of two well known phantoms were simulated with a pseudo ultrasound simulator as described in [12]. A 3 month old fetus is emulated in the second phantom image (Fig. 2(a)). The echogenicity map is drawn by constructing a bitmap image with different levels of scattering strength in the region of interest. Shepp-Logan phantom [10] simulates the slice of a human brain. The echogenicity map of the phantom contains ellipses with different absorption properties, which resembles the outline of a head. Clinical test suite constitutes real life US images collected from various US image databases. These test images

include slice of sagittal plane of cardiac 3D image (captured with 3D X3-1 matrix array transducer, Philips Healthcare, The Netherlands) [20]. A micro-calcified thyroid USG images are collected from open access thyroid US image database (TOSHIBA Nemio MX Ultrasound devices, Toshiba, Tokyo, Japan) [15]. In micro-calcified thyroid image, the calcified region gives a hyper-echogenic texture. The longitudinal section of carotid artery ultrasound images were taken from US carotid artery detection dataset (Sonix OP ultrasound scanner, Ultrasonix, Ohio, US and ATL HDI-3000 ultrasound scanner Advanced Technology Laboratories, Seattle, USA) [14, 18].

3.2 Result and Analysis

Experiments are done in two phases. In the first phase, phantom images corrupted with speckle noise are denoised using the proposed DsHSIWTV and other state-of-the-art approaches. The denoised images of a noisy phantom obtained from these approaches are shown in Fig. 2 and the zoomed view of the same are shown in Fig. 3. In the second phase, the de-speckling approaches are applied on real ultrasound images. Figure 4 shows the restored images obtained from different de-speckling techniques. Quantitative measures like SSIM & FOM (for analysing phantom images) and SSI & ENL (for analysing of real ultrasound images), computed from different denoised ultrasound images (obtained using different algorithms) are reported in Tables 1 and 2, respectively. Entries in bold denote the best result. It is noteworthy that two different kind of measures are used for reporting the quantitative results. This is done due to the fact that the ground truth echogenic maps are not available for the real ultrasound images. From Fig. 2, it is seen that OSRAD and DsHSIWTV provide better results as

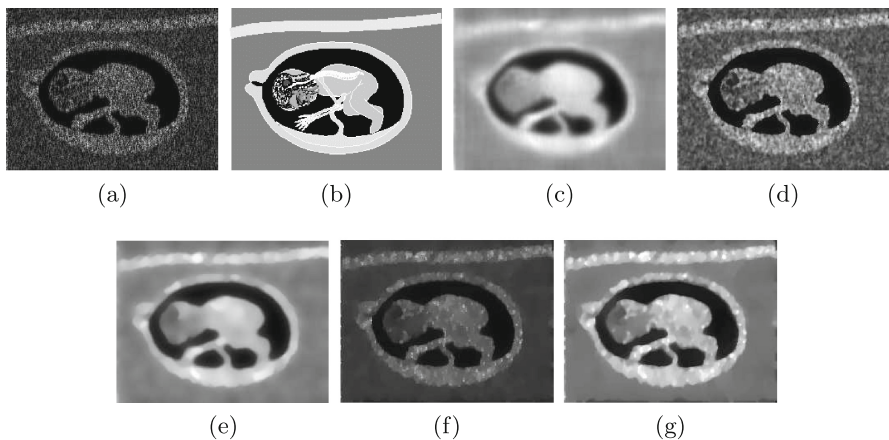


Fig. 2. (a) Noisy image of phantom with speckle variance $\sigma = 0.01$, (b) Echogenicity map, de-noised results using: (c) OBNLM, (d) ADMMS, (e) OSRAD, (f) RayleighTV, and (g) DsHSIWTV

compared to other approaches both in terms of smoothing and edge preservation. However, DsHSIWTV does not induce blurring effect near structural boundary and preserves finer details better in images in contrast to OSRAD. Rayleigh TV based method (Fig. 2(f)) also attains comparable results; however, some spike noise is noticed in the restored images. Zoomed view of the results, obtained using proposed approach and Rayleigh TV based method (Fig. 3), confirms this findings. Moreover, it is observed that due to use of TV as regularizer in RayleighTV, a staircase artifact and loss of directional information is noticed in the restored image (Fig. 3(a)) as opposed to the one obtained using proposed approach. Analysis of quantitative results (Table 1) also corroborates this fact. Among all the methods used, the proposed approach has the largest SSIM and FoM value for restored images. Both from quantitative and qualitative analysis, it can be seen that ADMMS has poor performance. This is due to the fact that the said method explicitly inhibits smoothing in the hyper-echogenic echotextured region.

Table 1. Values of Structural Similarity index (SSIM) and Figure of Merit (FoM) measures computed on de-noised results of phantom images obtained using various algorithms

Algorithm used	Values of different measures using			
	Fetus (Phantom 1)		Brain (Phantom 2)	
	SSIM	FoM	SSIM	FoM
ADMMS	0.9937	0.2583	0.9980	0.4633
OSRAD	0.9980	0.6504	0.9984	0.7769
OBNLM	0.9960	0.7639	0.9980	0.7533
RayleighTV	0.9829	0.6092	0.9955	0.5199
DsHSIWTV	0.9989	0.7445	0.9990	0.8046

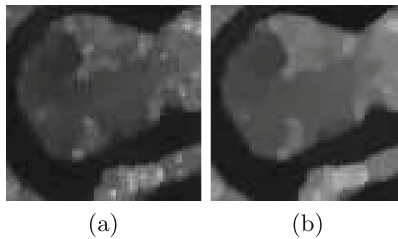


Fig. 3. Zoomed view of fetus phantom result obtained from (a) RayleighTV and (b) DsHSIWTV

Efficiency of the proposed approach over other strategies of denoising has also been confirmed by experimenting with clinical ultrasound images. This is

evident after observing the sample result as shown Fig. 4. Texture boundary and fine edges (Fig. 4(e and f)) in the images are well preserved by Rayleigh TV and proposed approach as opposed to others. Directional information is prominent in case of DsHSIWTV. Moreover, spike noise is produced with RayleighTV. OSRAD also preserves the strong edges but fails to preserve finer details and vague texture boundary leading to over-smoothing of image (Fig. 4(d)). From quantitative results (Table 2) it is found that the proposed approach has better speckle suppression index in almost all images except for carotid artery image. It is to be noted that though the output obtained using RayleighTV has good edge preservation (Fig. 4(e)), quantitative result is not promising. This is due to the isotropic nature of the regularizer that fails to differentiate the variations due to edge and those due to noise. While, the proposed approach uses anisotropy of the edge to measure the variation.

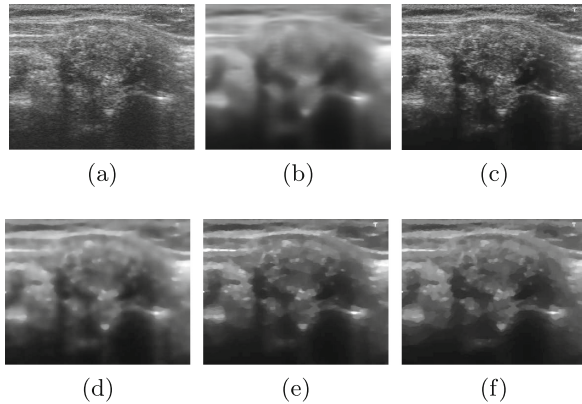


Fig. 4. (a) Noisy ultrasound image, De-noised results of micro-calcified thyroid ultrasound image obtained using: (b) OBNLM, (c) ADMMS, (d) OSRAD, (e) RayleighTV, and (f) DsHSIWTV

To summarize the result, the proposed approach has an edge as compared to other methods and provides a balance between smoothing and structural details preservation. The use of directional information of edges in the regularizer helps to induce the anisotropy while computing the variational energy of images. The degree of preservation of historical edges depending on nature of the echo texture helps to provide an implicit semantics coding of structures. Depending on this implicit information, smoothing could be controlled at the vicinity of the structures. Furthermore, the use of fuzzy clustering helped in capturing the vagueness of the echo texture due to which a proper degree of preservation of structural information could be defined. This in turn helped in identifying and removing the false variation due to speckle. Thus, integrating all this together in weights helps in preserving the structures in the restored images.

4 Conclusion

In this article, a speckle denoising strategy for clinical ultrasound images is proposed. The goal of the algorithm is to obtain a piecewise smooth restored ultrasound image while preserving fine structural details. A Bayesian MAP estimation framework based optimization function is minimized to obtain the restored images. To preserve the direction information of the edges and differential tissue characteristics in images, a weighted total variation based regularizer is used. The weights are recomputed from the present orientation information of the edges as well as previous structural information which were preserved over time.

Experiments are carried on various *in-silico* and *in-vivo* clinical ultrasound images to confirm the effectiveness of the proposed strategy. From the results it is found that the proposed approach has better speckle suppression ability, and the restored images are found to be piecewise smooth and highly contrasted. The structural details are also well preserved in the restored images, though vagueness in the echotexture in depth scan images makes the estimation of tissue related information difficult. Exploration of spatial information may improve the estimation of tissue information.

Table 2. Values of speckle suppression index (SSI) and effective number of look ups (ENL) computed from restored images of various clinical US images

Algorithm used	Values of different measures using							
	Sagittal slice		Longitudinal carotid		Microcalcified thyroid		Prostrate	
	SSI	ENL	SSI	ENL	SSI	ENL	SSI	ENL
ADMMS	0.7833 (0.0250)	7.9197 (0.6642)	0.5233 (0.0208)	5.0392 (0.3262)	0.8033 (0.0244)	6.0865 (0.3853)	0.445909 (0.0216)	7.007343 (0.0366)
OSRAD	0.6535 (0.0193)	9.4629 (0.8048)	0.5357 (0.0217)	4.1394 (0.3089)	0.5802 (0.0242)	8.5756 (0.5527)	0.495866 (0.0141)	6.460714 (0.5249)
OBNLM	0.5869 (0.0169)	8.9757 (0.7919)	0.4593(0.0257)	5.3135 (0.4489)	0.6055 (0.0320)	7.5775 (0.5502)	0.479192 (0.0172)	7.945573 (0.8990)
Rayleigh TV	0.7872 (0.0274)	8.1895 (0.8733)	0.6065 (0.0253)	3.9604 (0.3086)	0.7164 (0.0233)	6.4428 (0.4092)	0.7995 (0.0158)	3.6595 (1.2613)
D _s HSIW TV	0.5655(0.0213)	9.6198(1.3770)	0.5534(0.0232)	4.8645(0.4219)	0.4635(0.02490)	9.0935(0.3907)	0.3227(0.0166)	14.8395(3.5979)

References

1. Aubert, G., Aujol, J.F.: A variational approach to remove multiplicative noise. *SIAM J. Appl. Math.* **68**(4), 925–946 (2008)
2. Biucas-Dias, J., Figueiredo, M.: Multiplicative noise removal using variable splitting and constrained optimization. *IEEE Trans. Image Process.* **19**(7), 1720–1730 (2010)
3. Coupé, P., Hellier, P., Kervrann, C., Barillot, C.: Nonlocal means-based speckle filtering for ultrasound images. *IEEE Trans. Image Process.* **18**(10), 2221–2229 (2009)
4. Dellepiane, S.G., Angiati, E.: Quality assessment of despeckled SAR images. *IEEE J. Sel. Topics Appl. Earth Observations Remote Sens.* **7**(2), 691–707 (2014)

5. Dutt, V., Greenleaf, J.F.: Adaptive speckle reduction filter for log-compressed B-scan images. *IEEE Trans. Med. Imag.* **15**(6), 802–813 (1996)
6. El Hamidi, A., Ménard, M., Lugiez, M., Ghannam, C.: Weighted and extended total variation for image restoration and decomposition. *Pattern Recogn.* **43**(4), 1564–1576 (2010)
7. Feng, W., Lei, H., Gao, Y.: Speckle reduction via higher order total variation approach. *IEEE Trans. Image Process.* **23**(4), 1831–1843 (2014)
8. Han, Y., Feng, X.C., Baciuc, G., Wang, W.W.: Nonconvex sparse regularizer based speckle noise removal. *Pattern Recogn.* **46**(3), 989–1001 (2013)
9. Huang, Y.M., Ng, M.K., Wen, Y.W.: A new total variation method for multiplicative noise removal. *SIAM J. Imag. Sci.* **2**(1), 20–40 (2009)
10. Jain, A.K.: *Fundamentals of Digital Image Processing*. Prentice-Hall Inc., Upper Saddle River (1989)
11. Kang, M., Kang, M., Jung, M.: Total generalized variation based denoising models for ultrasound images. *J. Sci. Comput.* **72**(1), 172–197 (2017)
12. Krissian, K., Westin, C.F., Kikinis, R., Vosburgh, K.G.: Oriented speckle reducing anisotropic diffusion. *IEEE Trans. Image Process.* **16**(5), 1412–1424 (2007)
13. Li, S.Z.: *Markov Random Field Modeling in Image Analysis*, 2nd edn. Springer Science & Business Media, London (2009)
14. Loizou, C.P., Pattichis, C.S.: Despeckle filtering of ultrasound images. In: Suri, J., Kathuria, C., Molinari, F. (eds.) *Atherosclerosis Disease Management*. Springer, New York (2011). https://doi.org/10.1007/978-1-4419-7222-4_7
15. Pedraza, L., Vargas, C., Narvaez, F., Duran, O., Munoz, E., Romero, E.: An open access thyroid ultrasound image database. In: *Proceeding of SPIE*, vol. 9287, pp. 92870W–92870W-6 (2015)
16. Ramos-Llordén, G., Vegas-Sánchez-Ferrero, G., Martín-Fernández, M., Alberola-López, C., Aja-Fernández, S.: Anisotropic diffusion filter with memory based on speckle statistics for ultrasound images. *IEEE Trans. Image Process.* **24**(1), 345–358 (2015)
17. Rangayyan, R.M.: *Biomedical Image Analysis*. CRC Press, Washington D.C. (2004)
18. Riha, K., Masek, J., Burget, R., Benes, R., Zavodna, E.: Novel method for localization of common carotid artery transverse section in ultrasound images using modified Viola-Jones detector. *Ultrasound Med. Biol.* **39**(10), 1887–1902 (2013)
19. Steidl, G., Teuber, T.: Removing multiplicative noise by Douglas-Rachford splitting methods. *J. Math. Imag. Vis.* **36**(2), 168–184 (2010)
20. Tobon-Gomez, C., De Craene, M., McLeod, K., Tautz, L., Shi, W., Hennemuth, A., Prakosa, A., Wang, H., Carr-White, G., Kapetanakis, S., et al.: Benchmarking framework for myocardial tracking and deformation algorithms: an open access database. *Med. Image Anal.* **17**(6), 632–648 (2013)
21. Wang, H., Banerjee, A.: Bregman alternating direction method of multipliers. In: *Advances in Neural Information Processing Systems*, pp. 2816–2824 (2014)
22. Wu, Y., Feng, X.: Speckle noise reduction via nonconvex high total variation approach. *Math. Probl. Eng.* **2015**, 1–11 (2015)
23. Yang, J., Fan, J., Ai, D., Wang, X., Zheng, Y., Tang, S., Wang, Y.: Local statistics and non-local mean filter for speckle noise reduction in medical ultrasound image. *Neurocomputing* **195**, 88–95 (2016)
24. Yu, C., Zhang, C., Xie, L.: A multiplicative Nakagami speckle reduction algorithm for ultrasound images. *Multidimension. Syst. Sig. Process.* **23**(4), 499–513 (2012)
25. Yu, Y., Acton, S.T.: Speckle reducing anisotropic diffusion. *IEEE Trans. Image Process.* **11**(11), 1260–1270 (2002)

Fingerprint Image Quality Assessment and Scoring

Ram Prakash Sharma^(✉) and Somnath Dey

Indian Institute of Technology, Indore, India
{phd1501201003,somnathd}@iiti.ac.in

Abstract. Fingerprint quality estimation is an essential step for eliminating poor quality fingerprint images which can degrade the recognition performance of automatic fingerprint identification system (AFIS). A quality assessment technique along with fingerprint quality score will enable AFIS system to make appropriate decision regarding rejecting the low quality image and recapture a better quality fingerprint image. In this paper, we propose an effective method for evaluating fingerprint image quality (dry, normal dry, good, normal wet and wet) on a local level (block-wise). Feature vector for evaluating fingerprint quality covers moisture, mean, variance, ridge valley area uniformity and ridge line count. Block-wise quality label is assigned through pattern classification based on these features. In addition to quality labels, our proposed method also provides a quality score for a fingerprint image. Manually labeled dry, normal dry, good, normal wet and wet quality blocks of FVC 2004 *DB1_a* dataset is used to create a classification model using decision tree classifier. Block classification accuracy of 95.20% is achieved. Further, the same classification model is utilized to compute overall quality score of a fingerprint image. It has been observed that the overall quality score is accurate according to the manually labeled fingerprint image and also through visual inspection.

Keywords: Fingerprint · Quality label · Fingerprint block quality
Quality score · Decision Tree · Biometrics

1 Introduction

An automatic fingerprint identification system (AFIS) is a biometric recognition methodology which utilizes digital imaging techniques for fingerprint acquisition, storage as well as identification with minimum human intervention. However, the performance of such systems relies on the quality of the acquired fingerprint sample. Poor quality images result in spurious features, which degrade the performance of the system. So, fingerprint image quality assessment algorithms would be beneficial to improve the performance of an AFIS. If the quality of a fingerprint image is poor due to environmental factors such as dryness, wetness, less or more pressure, presence of dirt then the quality assessment algorithms

will help to reject poor quality images during fingerprint acquisition. A better quality image can be obtained by asking user to adjust conditions of his fingertip, rather than enhancing a low quality image. Generally, quality control is considered as an essential component of automatic biometric system prior to matching stage. Quality control acts as a filter to remove low quality images which may degrade the recognition performance.

Many of on-going and past efforts have been made for estimating the quality of fingerprint images [1, 2]. Majority of them are focused towards assigning a graded quality index to fingerprint images, from high to low or good to bad quality [3–6] while other gives a quality score to fingerprint image [7, 8]. Identification of the reason (dryness, wetness) behind the deterioration of the quality is also important. This impairment information helps to obtain a better quality image by allowing user to adjust conditions of his fingertip. Quality (dry, wet, good) identification of fingerprint images can also be utilized for adaptive enhancement of fingerprint images. So, the main motive of our work is to design a quality assessment and scoring module using quality dependent features, which helps to improve the decision making capability (acceptance or rejection of fingerprint image) of AFIS and performance of the enhancement process.

In this paper, we propose two-tier quality quantification approach which gives quality label (dry, normal dry, good, normal wet and wet) to fingerprint blocks as well as determines the quality score of a fingerprint image using these quality labels. It is possible that some areas of the fingerprint exhibit thick ridges, some other areas exhibit faint ridges and yet another areas have more equally spaced ridge-valleys. Based on these properties different regions of a fingerprint image can be labeled with the different qualities. Our two-tier quality quantification approach first determines the quality labels for all fingerprint blocks using five features, namely, moisture, mean, variance, ridge valley area uniformity and ridge line count. Next, these labels are used to compute the overall quality score. Decision tree classifier has been utilized to classify the feature values into dry, normal dry, good, normal wet and wet quality blocks.

2 Related Work

A literature review of some of the eminent fingerprint quality assessment techniques is presented in this section. The existing fingerprint quality estimation methods are classified into three categories [1] namely, local, global and classifier based quality estimation methods. Local quality estimation methods divide the fingerprint image into non-overlapping blocks and estimate quality of these blocks using block-wise features like orientation certainty level (OCL), ridge frequency, ridge thickness etc. Global quality analysis methods use features of entire image for quality assessment and classifier based methods assign good or bad quality label to a fingerprint image based on local and global features. Lim et al. [7] examines the local and global structure of a fingerprint image, where local structure means the texture like pattern of the ridges and valleys while global structure represents the smooth flow of the ridges and valleys. For examining the local structure it computes OCL, ridge frequency, ridge thickness and

ridge-to-valley thickness. Global structure analysis is done with continuity (orientation) and uniformity (ridge valley structure) features from the entire image. An overall quality score is obtained by combining these local and global quality measures. Chen et al. [8] proposed two quality measures: ridge and valley clarity to determine overlapping region of the distribution of ridges and valleys and global orientation flow to test the smoothness of the orientation map of the fingerprint. An overall image quality score is computed by combining the clarity score with orientation score using suitable weights. Shen et al. [3] proposed a method based on gabor features to estimate fingerprint quality. In this approach, a fingerprint image is divided into N blocks of size $w \times w$. For each block, m gabor features with m different orientations and their standard deviations are calculated. Standard deviations of the gabor features are used to determine the quality of the block based on a predefined threshold. A classifier based method in [4], defines the quality as degree of separation between match and non-match distributions of a given fingerprint. It classifies the quality of a fingerprint image into five levels: poor, fair, good, very good, and excellent using features vector consisting clarity of ridges and valleys, size of the image and measure of number and quality of minutiae. A neural network has been trained based on the mapping between feature vector and the normalized score to predict the suitable quality level of a given fingerprint image. Another classification method for fingerprint quality based on neural network has been proposed in [5]. Effective area, energy concentration, spatial consistency and directional contrast features are used to classify fingerprint images as high or low quality. Further, a comparative study of other ongoing and past efforts for quality assessment of fingerprint images can be found in [1, 2].

A block based quality evaluation method is proposed in [6], which uses directional strength, sinusoidal local ridge/valley pattern, ridge/valley uniformity and core occurrences. It classifies the blocks in good and bad quality labels based on these features using a self organizing map, radial basis function neural network and naive based classifier. In [9], a method to estimate block quality of fingerprint images is proposed. This method uses gray value of fingerprint block as input to a back propagation neural network. Output of the neural network is a continuous variable between 0 and 1, which represents the block quality score. Some other block-wise fingerprint image quality estimation methods can be found in [10, 11]. In [12], various local and global features used for assessment and estimation of fingerprint quality have been reported. A brief literature review of different fingerprint quality assessment techniques using features mentioned in [12] are presented in [13].

3 Proposed Method

Proposed system works in two phases: (1) Block quality labeling and (2) Fingerprint quality score computation. Block diagram of phase 1 and phase 2 are given in Fig. 1(a) and (b), respectively. In the first phase, a block quality labeling model is built to assign a suitable quality label Q (dry, normal dry, good, normal

wet and wet) to a fingerprint block. While in the second phase, overall quality score for a fingerprint image is computed using quality labels of all foreground blocks obtained using trained classifier model of phase 1. We have considered the following quality labels in this work.

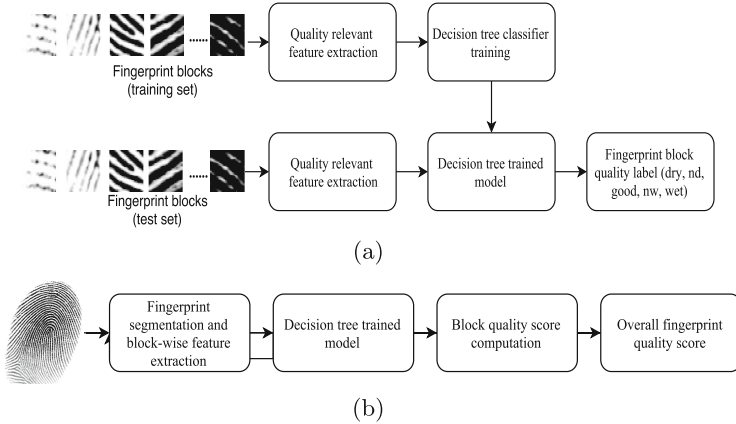


Fig. 1. Block diagrams of the proposed methods. (a) Block quality labeling and (b) Fingerprint quality score computation

- **Dry** blocks have fainted ridge-valley structure because of low pressure on the scanner surface or dryness of the skin as shown in Fig. 2(a). Large number of false minutiae can be detected in these blocks because of broken ridges.
- **Normal dry** quality blocks contain scratchy ridges whose thickness is less than the subsequent valley region in the block. Figure 2(b) shows some sample of normal dry quality blocks.
- **Good** quality blocks are those which have clearly separated ridges and valleys such that a minutiae extraction algorithm is able to operate very well as shown in Fig. 2(c).
- **Normal wet** quality blocks as shown in Fig. 2(d) have dark and hazy ridges with less valley region.
- **Wet** blocks are too dark that it is difficult to separate ridge-valley structure in them. Figure 2(e) shows some sample of wet quality blocks.

A fingerprint image is divided into blocks which can be of different quality labels and each quality labeled block should contribute certain score to obtain an overall quality score. As good quality blocks contain clear ridge/valley structure, their contribution should be the highest among normal dry, normal wet, dry and wet blocks in the overall quality score of a fingerprint. Normal dry and normal wet blocks have better ridge/valley representation than dry and wet blocks, respectively. Therefore, to obtain overall quality score we assign appropriate

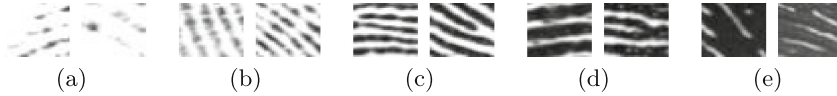


Fig. 2. Manually labeled blocks of different quality from FVC 2004 *Db1_a* database. (a) Dry (b) Normal dry (c) Good (d) Normal wet (e) Wet

quality weights to the different quality labels. A good quality block is assigned quality weight of 1 while normal dry and normal wet blocks are with 0.7, and dry and wet blocks are given 0.4 quality weights.

3.1 Block Quality Labeling

In block quality labeling, relevant quality features of each foreground block of a fingerprint image are computed. The foreground blocks of a fingerprint image are identified using variance based segmentation approach [14] with block-size of 32×32 . A decision tree classifier is modeled to map each foreground block of a fingerprint image into a quality label using extracted quality features.

3.1.1 Fingerprint Blocks (Training and Test Set)

We have collected 500 foreground blocks [14] of size 32×32 for each of the defined quality labels from the different fingerprint images of FVC 2004 *DB1_a*. Out of 500 blocks of each quality label, 400 blocks are used for training set and 100 blocks are used for test set.

3.1.2 Quality Relevant Feature Extraction

To cluster fingerprint blocks into defined quality bins we have proposed five relevant features which characterize dryness, wetness and good quality. We have identified five features namely, mean, variance, moisture, ridge valley area uniformity and ridge line count of a fingerprint image. All of these features are computed for a 32×32 sized foreground blocks of a fingerprint image. In a 500 dpi fingerprint image, a ridge valley pair is 8–12 pixel wide [15]. To accommodate at least two ridge valley pairs block size of 32×32 is selected. Features like moisture, ridge valley area uniformity and ridge line count are computed in a binary image, where the binary image is obtained using Otsu’s method [16]. Feature calculation for a block B is done as follows:

1. **Moisture (MI):** Moisture on a fingerprint will influence the dry or wet nature of fingerprint images. Unwanted ridge pixels in the valley region of fingerprint block are termed as moisture which is determined as the percentage of unwanted ridge pixels in the valley region of a fingerprint image. The percentage of unwanted ridge pixels are determined by subtracting percentage of average number of ridge pixels (λ) in a manually labeled good quality block from the percentage of total number of ridge pixels in that block.

The value of λ is obtained by averaging the percentage of ridge (black) pixels in 100 manually labeled good quality blocks. Computation of moisture (MI_b) for a foreground block B of size 32×32 is done using Eq. (1).

$$MI_B = \left(\frac{1}{|B|} \sum_{i=1}^{32} \sum_{j=1}^{32} B(i, j) [B(i, j) = 0] \right) * 100 - \lambda \quad (1)$$

$$\lambda = \left\{ \frac{\sum_{n=1}^{100} \left(\frac{1}{|B_{n_{good}}|} \sum_{i=1}^{32} \sum_{j=1}^{32} B_{n_{good}}(i, j) [B_{n_{good}}(i, j) = 0] \right)}{100} \right\} * 100 \quad (2)$$

Here, MI_B is the moisture level of block B , $B(i, j)$ is the value at pixel location (i, j) of binary block B , $|B|$ is the count of all pixels in the block ($32^2 = 1024$) and λ value is found as 51.25% using Eq. (2) where $B_{n_{good}}$ is manually labeled good quality block.

- ROI Mean (M):** Mean of foreground blocks of the fingerprint image shows overall gray level of the block and used as a quality predictor of a fingerprint block. Block-wise mean (M_B) of each of the foreground block B of size 32×32 is computed using Eq. (3).

$$M_B = \frac{1}{|B|} \sum_{i=1}^{32} \sum_{j=1}^{32} B(i, j) \quad (3)$$

- ROI Variance (V):** In order to obtain uniformity of a foreground block, block wise variance (V_B) is calculated using Eq. (4).

$$V_B = \frac{1}{|B|} \sum_{i=1}^{32} \sum_{j=1}^{32} (B(i, j) - M_B)^2 \quad (4)$$

- Ridge Valley Area Uniformity (RVAU):** A good quality fingerprint block should have almost equal regions of ridges and valleys. But in real fingerprint blocks it is found that these regions are not uniform across the different quality blocks. The ratio of ridge area versus valley area is calculated by determining the total number of ridge and valley pixels in the block. RVAU of a block ($RVAU_B$) is calculated using Eq. (5).

$$RVAU_B = \frac{\sum_{i=1}^{32} \sum_{j=1}^{32} B(i, j) [B(i, j) = 0]}{|B| - \sum_{i=1}^{32} \sum_{j=1}^{32} B(i, j) [B(i, j) = 0]} \quad (5)$$

- Ridge Line Count (RLC):** The number of ridge lines in a block represents different quality of a fingerprint block. It is desirable that RLC in foreground blocks of a fingerprint image will remain almost constant. However, this number varies largely across the different blocks of a fingerprint image as it may have the different quality labels. RLC is computed for blocks of size 32×32 from a ridge map with one pixel thickness, which is obtained by thinning

morphological operation. Ridge map is rotated vertically using orientation estimation method adapted from [4] to make the ridge lines vertical and count the pixels from white to black and black to white passing through each row of a block. This gives the count of twice the number of ridges present in each row of a block. The half of the maximum of these counts is considered as RLC of the block (RLC_B) which is defined in Eq. (6).

$$RLC_B = \frac{1}{2} \times \text{Max}_i \left(\sum_{j=1}^{j=32} B(i, j) [b- > w || w- > b] \right) \text{ for } i = 1 \text{ to } 32 \quad (6)$$

3.1.3 Decision Tree Classification

A five dimensional feature vector is extracted from each of the 2500 blocks (2000 for training set and 500 for testing set) as given in Subsect. 3.1.2. Feature vectors of 2000 training blocks are given as input to train a decision tree model for block quality labeling. Decision Tree (DT) is a non-parametric supervised learning method used for classification. Its goal is to create a classification model that predicts the target variable or class by learning simple decision rules inferred from the data features. The trained decision tree model is utilized to classify the 500 test set blocks (100 from each quality label) into the most suitable quality label. DT also gives a probability score (P_Q) to each classified block which represents similarity with the assigned quality label (Q).

3.2 Fingerprint Quality Score Computation

It this phase, weight associated with each quality label (found using phase 1 trained decision tree model) is used to compute the quality score of each block and these individual quality scores are utilized to compute the overall quality score of a fingerprint image.

3.2.1 Fingerprint Segmentation and Block-Wise Feature Extraction

The foreground blocks of a fingerprint image are identified using variance based segmentation approach [14] with block-size of 32×32 . Features from foreground blocks are extracted as given in Subsect. 3.1.2, to form a five dimensional feature vector for each block.

3.2.2 Decision Tree Trained Model

The trained DT model obtained in Subsect. 3.1.3 of phase 1 is utilized for quality labeling of foreground blocks of a fingerprint image. It also gives a probability (P_Q) of similarity with the assigned quality label (Q).

3.2.3 Block Quality Score Computation

The quality label and probability score assigned to each foreground block of a fingerprint image is utilized to compute the block quality score. Quality weights

associated with block quality label is multiplied with its probability score to obtain the quality score for a particular block. Contributions of the different quality labels Q (dry, normal dry, good, normal wet and wet) in the overall quality score are obtained by summation of block quality score of each quality label as given in Eq. (7).

$$S_Q = \frac{1}{N_Q} \sum_{i=1}^{i=N_Q} W_Q \times P_Q(i) \quad (7)$$

Here, S_Q denotes the contribution of quality label Q in the overall quality score and W_Q represents the block quality weights, which are 0.4, 0.7, 1, 0.7 and 0.4 for dry, normal dry, good, normal wet and wet quality labels, respectively. N_Q and P_Q are number and probability score of blocks labeled with quality Q .

3.2.4 Overall Quality Score

The overall quality score of a fingerprint image is obtained by summing up the contributions of each defined quality label Q . The overall quality score is computed using Eq. (8).

$$QS = \frac{1}{5} \sum_Q S_Q \times 100 \quad (8)$$

4 Experimental Results

To assess the efficiency of the proposed block quality prediction method, a number of experiments have been performed on fingerprint blocks of FVC 2004 *DB1.a* dataset which contains varying quality fingerprint images. The dataset was acquired using optical sensor and contains 8 impressions of 100 fingers. Each of the fingerprint image is of 500 dpi with 640×480 size. First, the DT classification model is tested with 500 blocks of different quality labels (100 blocks from each quality label). The results of block quality prediction are presented

Table 1. Results of block quality labeling.

		Decision tree classification					
		<i>Dry</i>	<i>Normal Dry</i>	<i>Good</i>	<i>Normal Wet</i>	<i>Wet</i>	<i>Total</i>
Subjective Quality	<i>Dry</i>	98	2	0	0	0	100
	<i>Normal Dry</i>	1	95	4	0	0	100
	<i>Good</i>	0	4	93	3	0	100
	<i>Normal Wet</i>	0	0	5	93	2	100
	<i>Wet</i>	0	0	1	2	97	100
	<i>Total</i>	99	101	103	98	99	500
Accuracy		98.00%	95.00%	93.00%	93.00%	97.00%	

in Table 1. Results obtained from DT classifier are compared with the manually labeled blocks of the different qualities. DT classifier predicts 99 blocks in dry quality bin, out of which 98 are predicted correctly and 1 is predicted in wrong quality bin. So, accuracy of prediction of dry blocks is $98/100 \times 100 = 98.00\%$. Similarly, number of correctly classified blocks for normal dry, good, normal wet and wet are 95, 93, 93 and 97, which give accuracy of 95.00%, 93.00%, 93.00% and 97.00% respectively. Results show the accuracies of prediction of the correct quality cluster for dry and wet blocks are very good. Overall accuracy of the DT classifier is calculated using Eq. (9).

$$\begin{aligned} Accuracy &= \frac{T_{blocks} - F_{blocks}}{T_{blocks}} \times 100 \\ &= \frac{500 - 24}{500} \times 100 \\ &= 95.20\% \end{aligned} \quad (9)$$

Here, T_{blocks} is the total fingerprint blocks in the test dataset and F_{blocks} is number of wrongly classified with respect to manual labeling.

Results of the proposed quality labeling is compared with block quality analysis method proposed by Lim et al. in [6]. Lim's method classifies the fingerprint blocks into two quality class (good and bad). The proposed quality labels are divided into two classes namely, good and bad to make our block quality labeling method comparable with Lim's method. The dry, normal dry, normal wet and wet quality blocks are considered as bad quality class while good quality blocks remain as good class. Comparison results in Table 2 show that the DT classifier model with the proposed features outperforms the self organising map (SOM), radial basis function neural network (RBFNN) and naive bayes classifier with directional strength, sinusoidal local ridge/valley pattern, ridge/valley uniformity and core occurrences features. The good and bad quality blocks are predicted with accuracy of 93% and 95.75%, respectively with our proposed approach, which are the highest for the good and bad quality blocks prediction among the different classifier models.

Table 2. Comparison of the proposed block quality labeling with Self Organising Map (SOM), Radial Basis Function Neural Network (RBFNN) and Naive Bayes classifier proposed in [6]

Subjective Quality	SOM		RBFNN		Naive Bayes		Proposed DT	
	Good	Bad	Good	Bad	Good	Bad	Good	Bad
Good (100)	86 (86%)	14 (14%)	91 (91%)	9 (9%)	89 (89%)	11 (11%)	93 (93%)	7 (7%)
Bad (400)	43 (10.75%)	357 (89.25)	20 (5%)	380 (95%)	28 (7%)	372 (93%)	17 (4.25%)	383 (95.75%)

To evaluate the performance of the proposed quality scoring method, quality scores of the 100 manually labeled fingerprint images of each quality (dry, normal dry, good, normal wet and wet) from FVC2004 *Db1_a* are computed.

Figure 3 shows fingerprint images of five different qualities and their corresponding quality maps along with their quality score. Quality scores given by the proposed quality scoring method for the fingerprint images of quality (Q) are compared to the scores of NFIQ 2.0 [17]. Table 3 shows the mean of the quality scores given by the proposed quality scoring method and NFIQ 2.0 method for the fingerprint images of each quality label Q. Fingerprint images of different qualities are well separated using the proposed quality scoring method (normal dry:50.28, good:79.91 and normal wet:47.53) as compared to the quality scores (normal dry: 62.30, good:68.82 and normal wet:62.80) of NFIQ 2.0. The good quality fingerprint images are getting higher mean score (79.91) using the proposed method as compared to the mean score (68.82) of NFIQ 2.0. Similarly, mean scores of other quality labels are compared which show the quality scores given by the proposed method are more accurate.

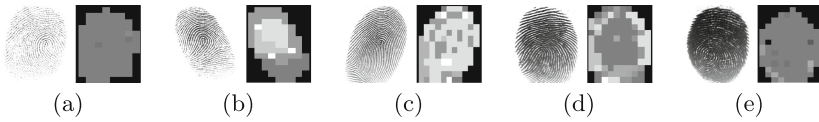


Fig. 3. Fingerprint images of different qualities from FVC 2004 *Db1.a* database and their corresponding quality maps. (a) Dry (QS:39.95) (b) Normal dry (QS:49.17) (c) Good (QS:78.14) (d) Normal wet (QS:47.74) (e) Wet (QS:36.13). Blocks with brighter color in quality maps indicate higher quality region

Table 3. Average quality scores given by NFIQ 2.0 [17] and the proposed method for manually labeled fingerprint images of dry, normal dry, good, normal wet and wet quality (100 each)

	Dry	Normal Dry	Good	Normal Wet	Wet
NFIQ 2.0	49.55	62.30	68.82	62.8	50.08
Proposed method	37.12	50.28	79.91	47.53	34.58

5 Conclusions

This paper proposes a method to assess fingerprint image quality on a local level using DT classifier. Five dimensional quality relevant feature vector has been proposed to build the DT model. Feature vector includes moisture, mean, standard deviation, RVAU and RLC. DT model gives an accuracy of 95.20% to predict the correct quality labels from the blocks of the different qualities. The benefits of the proposed method are in two folds. The proposed method divides the fingerprint image into the different quality regions using DT model. This result can be utilized to enhance the fingerprint image using local enhancement techniques. Further, a poor quality fingerprint image can be rejected based on

the overall quality score and a new fingerprint impression can be recaptured by asking users to provide his fingerprint impression for reacquisition. The proposed method has been tested on the manually labeled FVC 2004 *Db1_a* dataset for dry, normal dry, good, normal wet and wet images. Result shows accurate estimation of the quality scores of low (dry, wet), medium (normal dry and normal wet)) and high (good) quality fingerprint images.

Acknowledgment. The authors are thankful to Science and Engineering Research Board (SERB), DST (ECR/2017/000027), Govt. of India for providing financial support to carry out this research work.

References

1. Alonso-Fernandez, F., Fierrez, J., Ortega-Garcia, J., Gonzalez-Rodriguez, J., Fronthaler, H., Kollreider, K., Bigun, J.: A comparative study of fingerprint image-quality estimation methods. *IEEE Trans. Inf. Forensics Secur.* **2**(4), 734–743 (2007)
2. Bharadwaj, S., Vatsa, M., Singh, R.: Biometric quality: a review of fingerprint, iris, and face. *EURASIP J. Image Video Process.* **2014**(1), 34 (2014)
3. Shen, L.L., Kot, A., Koo, W.M.: Quality measures of fingerprint images. In: Bigun, J., Smeraldi, F. (eds.) *AVBPA 2001*. LNCS, vol. 2091, pp. 266–271. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45344-X_39
4. Tabassi, E., Wilson, C.L.: A novel approach to fingerprint image quality. In: *IEEE International Conference on Image Processing*, vol. 2, pp. 37–40. IEEE (2005)
5. Yang, X.K., Luo, Y.: A classification method of fingerprint quality based on neural network. In: *2011 International Conference on Multimedia Technology*, pp. 20–23 (2011)
6. Lim, E., Toh, K.A., Suganthan, P.N., Jiang, X., Yau, W.Y.: Fingerprint image quality analysis. In: *International Conference on Image Processing*, vol. 2, pp. 1241–1244 (2004)
7. Lim, E., Jiang, X., Yau, W.: Fingerprint quality and validity analysis. In: *Proceedings of International Conference on Image Processing*, vol. 1, pp. 469–472 (2002)
8. Chen, T.P., Jiang, X., Yau, W.Y.: Fingerprint image quality analysis. In: *International Conference on Image Processing*, vol. 2, pp. 1253–1256 (2004)
9. Xie, R., Qi, J.: Continuous fingerprint image quality estimation based on neural network. In: *2010 International Symposium on Intelligent Signal Processing and Communication Systems*, pp. 1–4 (2010)
10. Xie, S.J., Yoon, S., Yang, J.C., Park, D.S.: Rule-based fingerprint quality estimation system using the optimal orientation certainty level approach. In: *2nd International Conference on Biomedical Engineering and Informatics*, pp. 1–5 (2009)
11. Awasthi, A., Venkataramani, K., Nandini, A.: Image quality quantification for fingerprints using quality-impairment assessment. In: *2013 IEEE Workshop on Applications of Computer Vision*, pp. 296–302 (2013)
12. Olsen, M.A., Smida, V., Busch, C.: Finger image quality assessment features 2013 definitions and evaluation. *IET Biom.* **5**(2), 47–64 (2016)
13. Yao, Z., Bars, J.M.L., Charrier, C., Rosenberger, C.: Literature review of fingerprint quality assessment and its evaluation. *IET Biom.* **5**(3), 243–251 (2016)
14. Mehtre, B.M.: Fingerprint image analysis for automatic identification. *Mach. Vis. Appl.* **6**(2), 124–139 (1993)

15. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of Fingerprint Recognition, 2nd edn. Springer Publishing Company, Incorporated, Heidelberg (2009). <https://doi.org/10.1007/978-1-84882-254-2>
16. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979)
17. Tabassi, E.: Development of NFIQ 2.0. NIST (2015). <https://www.nist.gov/services-resources/software/development-nfiq-20>

A Multi-objective Evolutionary Algorithm for Color Image Segmentation

Kazi Shah Nawaz Ripon^{1(✉)}, Lasker Ershad Ali², Sarfaraz Newaz³,
and Jinwen Ma²

¹ Department of Computer Science, Norwegian University of Science
and Technology, Trondheim, Norway

ksripo@ntnu.no

² Department of Information Science, Peking University, Beijing, China

³ Department of CSE, Bangladesh University of Engineering and Technology,
Dhaka, Bangladesh

Abstract. In this paper, we present a multi-objective segmentation approach for color images. Three objectives, *overall deviation*, *edge value*, and *connectivity measure*, are optimized simultaneously using a multi-objective evolutionary algorithm (MOEA). To demonstrate the effectiveness of the proposed approach, experiments are conducted on benchmark images. The results justify that the proposed approach is able to partition color images in a number of segments consistent with human visual perception. For quantitative evaluation, we extend the existing *Probabilistic Rand Index* (PRI) considering multi-objective segmentation. The outcomes show that the proposed approach can obtain non-dominated and near-optimal segment solutions satisfying several criteria simultaneously. It can also find the correct number of segments automatically.

Keywords: Image segmentation · Multi-objective evolutionary optimization · Probabilistic Rand Index (PRI) · Color image · Pareto-front

1 Introduction

Images are considered one of the most important mediums of conveying information. Understanding images and extracting the information from them is an important aspect of many practical applications in various fields such as biology, medicine, remote sensing, chemistry, robotics, and industry. Image segmentation is one of the most significant and basic tasks in the field of image processing and recognition. The main goal of image segmentation is to partition an image into multiple non-intersecting regions (set of pixels) having high similarity among the pixels within a region, while the pixels among neighboring regions are significantly dissimilar with respect to some similarity measures. A large variety of different segmentation approaches have been proposed for monochrome and color images [2, 3, 5]. However, color image segmentation techniques are considered more appealing since they can provide more information than grey level

images, and the human eye is able to better detect objects when color is available within the image [3].

Real-world image segmentation problems actually require considering multiple objectives, i.e., minimize overall deviation, minimize segment overlap, maximize connectivity, minimize the number of features, or minimize the error rate of the classifier. However, existing image segmentation approaches are generally concerned with a single objective [4, 11]. By contrast, practical segmentation problems are multi-objective by nature and they require the decision makers to consider a number of criteria before arriving at any conclusion. A segmentation that is optimal with respect to a given criterion might be a poor candidate for some other criteria. Thus, a single solution that can optimize all objectives simultaneously does not necessarily exist. Hence, the trade-offs (Pareto-optimality) involved in considering several different criteria provide useful insights for the decision makers. Consequently, image segmentation falls into the category of multi-objective optimization problems.

To date, relatively few techniques have been developed for multi-objective image segmentation [9, 13]. Most of these algorithms suffer from the “cluster number dependency” problem, where the user should provide an accurate number of clusters in advance [12]. However, in most practical situations, it is not known in advance. In addition, none of the proposed approaches considers the use of Pareto-optimality. The case is far worse in the case of color images, where there exists no approach for segmentation considering multiple objectives.

Unlike conventional methods that aggregate multiple objectives to form a composite scalar objective, multi-objective evolutionary algorithms (MOEAs) are capable of considering each objective separately and guiding the search to discover the global Pareto-optimal front. Motivated by this, in this paper, we propose a multi-objective segmentation approach for color images with the use of Pareto-front by simultaneous optimization of three objectives. The objectives, (i) overall deviation, (ii) edge value, and (iii) connectivity measure, are simultaneously optimized using the Strength Pareto Evolutionary Algorithm-2 (SPEA-2) [15]. Experiments on ten color images from the Berkeley Image Segmentation Dataset (BSDS300) [8] show that our proposed approach is able to partition natural and human scenes in meaningful objects. Experimental results also justify that our approach can find a set of non-dominated and near-optimal segmentation by simultaneous optimization of multiple objectives. This is particularly beneficial to decision makers, as they can select the best compromise solution according to specific segmentation objectives required in different cases.

We also present quantitative evaluation based on the well known concept of Probabilistic Rand Index (PRI) [14]. The original PRI was formulated for image segmentation based on a single objective, where the approach under consideration produces only one segmentation solution. Since we are using MOEA, instead of a single output image solution, we will find a set of Pareto-optimal segmentation solutions. Therefore, we modify the original PRI for handling a set of final solutions to produce the final PRI value. Based on this *Modified PRI*, we compare our approach with the algorithms proposed by Amelio and Pizzuti [1]

(single objective and considers only the gray-level information), by Maji et al. [7] (single objective, but for color images), and by Amelio and Pizzuti [2] (single objective, but for color images). Comparison shows that the proposed approach gets better segmentation accuracy for most of the test images.

The rest of this paper is organized as follows. Section 2 explains the proposed approach. Section 3 describes the *Modified PRI* as the proposed quantitative evaluation criteria for color image segmentation. Section 4 provides the experimental results and discusses the findings. Section 5 concludes the paper with suggestions for future research.

2 Proposed Approach

Our approach can be summarized in the following steps: (i) representation of the input image, (ii) generation of a minimum spanning tree (MST) from this, (iii) initial segmentation from the MST, and (iv) utilizing the MOEA to optimize the objectives and to produce the final set of Pareto-optimal segmented images.

2.1 Representation of Individuals

The representation of individuals is a graph structure [13]. Figure 1 illustrates the genotype and phenotype for a 4×4 pixel image. In this figure, each number represents an index in the genotype which corresponds to the pixel index in the two-dimensional input image. The dotted sections in the phenotype indicate the segmentation. The length of a genotype is equal to the number of pixels of the input image. Each gene contains one out of five possible values; $\{left, right, up, down, none\}$; that describes how the graph node representing the input image pixel at the index of that gene is connected to its neighbors. Each graph node can connect to either one of its four cardinal neighbors, or to itself. If a graph node at an edge of the image plane points in an outwards direction, it is treated as having the value none. This means that all possible chromosome permutations are valid.

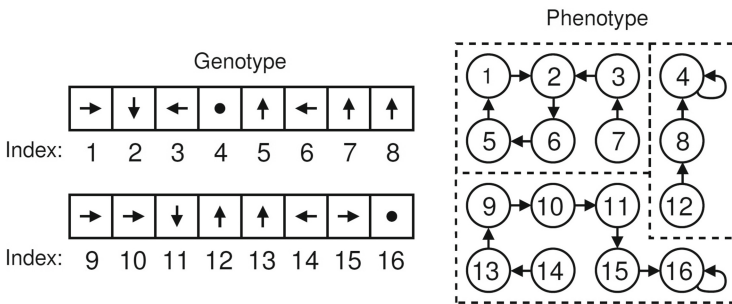


Fig. 1. Individual representation for a 4×4 pixels image.

Initial genotype sequences are generated by constructing a MST from the input image. This is to provide a good starting point for image segmentation. The input image is treated as a graph where each pixel is a node that is connected to each of its cardinal neighbors. The weight of each edge is given by the *Euclidean* distance in RGB color space between the two neighbors. From the initial image graph, we utilize Prim's algorithm [6] to generate a MST from a random starting point. Since a random starting point is used, a different MST is generated as the basis of the genotype for each initial individual.

2.2 Generation of Initial Segments

In order to evaluate an individual, the genotype is required to be converted into the phenotype (initial segmentation). For this, the directions of edges in the graph described by the genotype is ignored. Starting with the first pixel node in the graph, all directly or indirectly connected pixel nodes are assigned to the same segment. This process is continued until all pixel nodes have been assigned to a segment.

2.3 Objective Functions

After creating the initial segmentation, three objectives are simultaneously optimized using the SPEA-2. The first objective, the *overall deviation*, is a measure of the similarity of pixels in the same segment, as defined in Eq. 1:

$$\text{overall-deviation}(C) = \sum_{C_k \in C} \sum_{i \in C_k} \text{dist}(i, \mu_k) \quad (1)$$

where C is the set of all segments, μ_k is the centroid of the pixels in the segment C_k , and $\text{dist}()$ is the distance function. Overall deviation should be minimized. Minimizing overall deviation roughly increases the number of segments. The distance function, $\text{dist}()$, is the *Euclidean* distance in the RGB color space, and is defined as:

$$\delta_{RGB} = \sqrt{\Delta R^2 + \Delta G^2 + \Delta B^2} \quad (2)$$

The second objective, the *edge value*, evaluates the overall summed distances on boundaries between the segments. This value is a measure of the difference in the boundary among the segments. This objective should be maximized. However, to keep similarity with other two objectives, we convert it as subject to minimization by negating it as shown in Eq. 3. Here, N is the number of pixels, F_i indicates the 4 nearest neighbors of pixel i .

$$\text{Edge}(C) = - \sum_{i=1}^N \left(\sum_{j \in F_i} x_{i,j} \right), \quad (3)$$

$$\text{where, } x_{c,s} = \begin{cases} \text{dist}(c, s) & \text{if } \nexists C_k : c, s \in C_k \\ 0, & \text{otherwise} \end{cases}$$

The third objective, the *connectivity measure*, is defined in Eq. 4. This objective evaluates the degree to which neighboring pixels have been placed in the same segment, as follows:

$$Conn(C) = \sum_{i=1}^N \left(\sum_{j=1}^L x_{i,nn(j)} \right), \quad (4)$$

$$\text{where, } x_{c,s} = \begin{cases} \frac{1}{j} & \text{if } \nexists C_k : c, s \in C_k \\ 0 & \text{otherwise} \end{cases}$$

Here, N is the number of pixels in a segment, $nn(j)$ is the j -th nearest neighbour of the i -th pixel, L is a parameter determining the number of neighbors that contribute to the connectivity measure. In this work, we use $L = 8$. As an objective, the connectivity measure will also be minimized.

2.4 Evolutionary Operators

We use a tournament selection of size 4 in our experiments. Simple uniform crossover operator combines two randomly selected parent individuals to produce two child individuals. When applied, the mutation operator selects a random gene in a parent individual and sets it to a new value which is randomly selected from $\{\textit{left}, \textit{right}, \textit{up}, \textit{down}, \textit{none}\}$.

3 Evaluation Criterion: *Modified PRI*

The Berkeley dataset contains multiple human-traced segmentation for each color image, all of those are considered equally reliable. Therefore, the comparison should be made against all the manually obtained ground-truth segmentations. For such comparison, *Probabilistic Rand Index* (PRI) is introduced in [14] as an extension of *Rand Index* [10] which was designed to assess clustering methods. However, PRI is designed to evaluate the segmentation approaches those produce single final segmented solution only.

On the contrary, our aim to find a set of Pareto-optimal segmented outputs instead of a single output image by simultaneous optimization of three objectives. Therefore, we have modified the PRI into *Modified PRI* to asses multiple trade-off solutions. Given a set $\{GT_1, \dots, GT_T\}$ of ground-truth segmentations of an image I consisting of n pixels, and a test set of Pareto-optimal segmentation results $\{I_1, \dots, I_p\}$, the *Modified PRI* is defined as:

$$\begin{aligned} \textit{Modified PRI} (\{I_1, \dots, I_p\}, \{GT_1, \dots, GT_T\}) \\ = \frac{1}{H} \sum_{i \neq j} [c_{ij} p_{ij} + (1 - c_{ij})(1 - p_{ij})] \end{aligned} \quad (5)$$

where c_{ij} denotes the event that pixels i and j have the same label, p_{ij} is the probability, and $H = n \times (n - 1)/2$ is the total number of pixel pairs. Similar to the PRI, the *Modified PRI* values also varies between 0 and 1, where 0 means that $\{GT_1, \dots, GT_T\}$ and $\{I_1, \dots, I_p\}$ are completely dissimilar.

4 Experiments and Results

In this section, we present the results of our proposed approach on ten test images from the BSDS300 [8] and compare the performances in partitioning natural and human scenes in meaningful objects with the segmentations obtained by *C-GeNCut* [2], and *Biased NCut* [7] (referred as *C-NCut* hereafter) both for color images, and by *GeNCut* [1] that takes into account only gray-scale information, on the same images.

4.1 Experimental Setup

The parameters for the SPEA-2 are: population size = 50; generations = 100; archive size = 20; crossover probability = 0.7; mutation probability = 0.2. We use a constraint on initial segment size within the range of 1 to 50. In our experiments, five independent runs are made with each test image and the final Pareto-fronts from these runs are combined. Finally, a non-dominated sorting is performed to constitute the best non-dominated set of solutions.

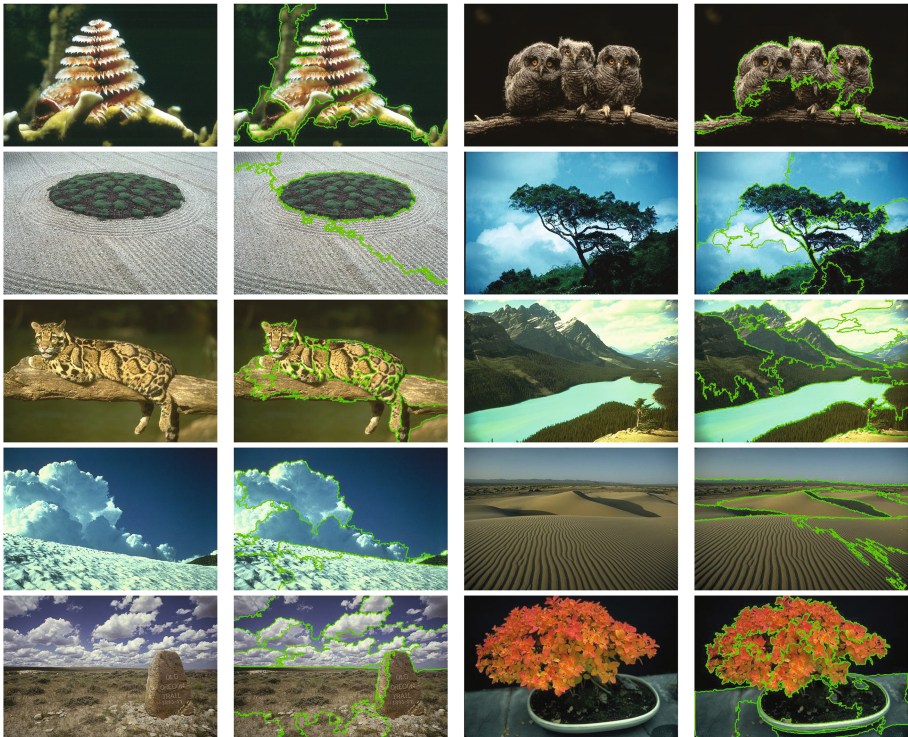


Fig. 2. Segmentation obtained by our proposed approach. For each image, the original version together with the segmentation results are presented.

4.2 Results and Discussion

Figure 2 presents the segmentation outputs produced by the proposed approach by depicting the contours of the regions on the original image. It is worthy to mention that each of these ten examples are one member of the final Pareto-optimal segmentation for each image. Each image of this figure is selected randomly from the corresponding Pareto-front. The figure also shows that the visual perception of the segmentation results is quite positive. For each test image, the main objects are identified as well as the segmentation process can successfully extract the most meaningful features.

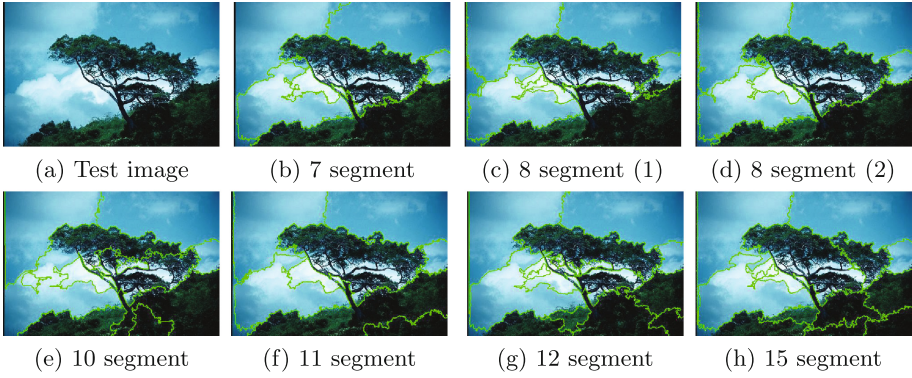


Fig. 3. Examples of the image segmentation results for image index 147091 with number of segments.

Another objective of this work is to verify whether various trade-off segmentations are obtained. Figure 3 shows examples of the segmentation results for image index 147091. The results show that several trade-off segmentations with different numbers and shapes, all of which can be considered to be relatively good from visual perspective. This also shows that our proposed approach can find the optimal/near-optimal number of segments automatically.

This is also justified by Fig. 4 which shows the obtained Pareto-front by the proposed approach. From the figure, it can be found that our approach can successfully optimize the objectives simultaneously and the obtained solutions of each image have different Pareto-fronts. Although the shapes of the Pareto-fronts are different, a wide range of solutions is found in all cases. The diverse solutions, in particular the extreme solutions, are useful for real-world scenarios where the decision maker can select the best compromising segmented solutions from the non-dominated set of solutions according to the specific requirements or scenarios. All these ultimately justify the effectiveness of our proposed approach as a multi-objective image segmentation approach for color images.

Table 1 presents the quantitative comparison of our proposed approach with *C-GeNCut* [2], *C-NCut* [7], and *GeNCut* [1]. This table is partially taken from [2].

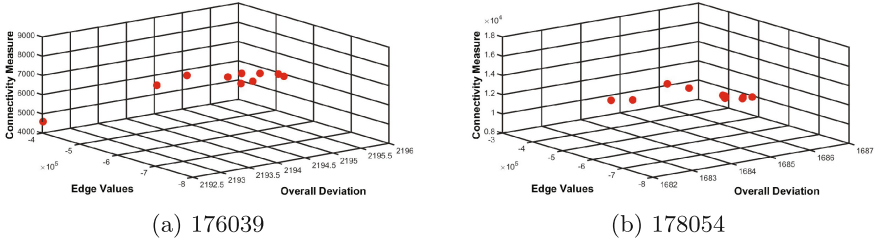


Fig. 4. Obtained Pareto solutions by the proposed method.

Table 1. Comparison based on Modified PRI

Image index	GeNCut	C-GeNCut	C-Ncut	Proposed Max	Approach Avg
I1 (12074)	0.7308 (0.0118)	0.782 (0.0101)	0.7512 (0.0018)	0.7424	0.7206 (0.0138)
I2 (42044)	0.8036 (0.0186)	0.8288 (0.0379)	0.7565 (0.0001)	0.8311	0.7973 (0.0128)
I3 (86016)	0.6443 (0.0637)	0.7526 (0.0263)	0.7862 (0.0003)	0.7946	0.7766 (0.0157)
I4 (147091)	0.7041 (0.0183)	0.7052 (0.0183)	0.6651 (0.0017)	0.7476	0.7314 (0.0175)
I5 (160068)	0.8215 (0.0002)	0.8361 (0.0163)	0.8217 (0.0001)	0.7475	0.7393 (0.02)
I6 (176035)	0.7797 (0.0375)	0.8361 (0.0075)	0.8557 (0.0001)	0.7919	0.7203 (0.0363)
I7 (176039)	0.7889 (0.0368)	0.8339 (0.0213)	0.826 (0.0001)	0.8341	0.7996 (0.0252)
I8 (178054)	0.7035 (0.0081)	0.7613 (0.0063)	0.7068 (0.0001)	0.7653	0.7622 (0.0023)
I9 (216066)	0.7425 (0.0059)	0.7653 (0.0076)	0.7399 (0.0001)	0.7719	0.7562 (0.0251)
I10 (353013)	0.8088 (0.0198)	0.8235 (0.0065)	0.8338 (0.0001)	0.755	0.74 (0.0211)

It is necessary mentioning that all the compared methods produce single output segmented solution. The first two methods are proposed for color images and the last one for gray-scale information. Whereas, our proposed approach produces a set of trade-off segment solutions. The values in bold face are the best and the values within braces are the standard deviation values. Based on *Modified PRI* as mentioned in the table, it is evident that our proposed approach finds the best *Modified PRI* for most of the test images (6 out of 10 images). Moreover,

for the other 4 images where our approach can not find the best values, the values obtained by our approach are still satisfactory. In short, the *Modified PRI* values show that our approach can find a number of segments equal to one of the ground-truth segmentations.

From this table, it can also be observed that in some cases the standard deviation values obtained by our approach are relatively large. Considering multi-objective evolutionary optimization, this phenomenon can be considered as “good”. It confirms an extra advantage of the proposed approach—its ability to find extreme solutions along the Pareto-front. Finding extreme solutions are, in particular, very important for MOEAs. This is because, the decision maker can select the best compromise solution according to specific segmentation objective required in different cases.

5 Conclusion

This paper presents a multi-objective segmentation approach for color images by optimizing three objectives simultaneously. To quantitatively assess multiple trade-off solutions in terms of test images with multiple ground-truth examples, we also extend an existing performance criteria into a modified performance index (*Modified PRI*). Experimental results justify that our proposed approach is able to segment color images in a number of regions that adhere well to the human visual perception. The quantitative evaluation also shows that our proposed approach is competitive with state-of-the-art methods for color image segmentation. In addition, our proposed approach is capable of searching a set of near-optimal trade-off segmentation solutions while finding the correct number of clusters automatically. This is essential for real-world segmentation as the trade-offs involved in considering several different criteria provide useful insights to decision makers by providing the flexibility to consider a number of criteria before choosing a solution. In the future, we would like to implement other segmentation criteria as optimization objectives to test the effectiveness of each. Implementing other recent MOEAs to analyze their behavior and performance could be another interesting avenue for future works.

References

1. Amelio, A., Pizzuti, C.: An evolutionary and graph-based method for image segmentation. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012. LNCS, vol. 7491, pp. 143–152. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32937-1_15
2. Amelio, A., Pizzuti, C.: A genetic algorithm for color image segmentation. In: Esparcia-Alcázar, A.I. (ed.) EvoApplications 2013. LNCS, vol. 7835, pp. 314–323. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37192-9_32
3. Cheng, H.D., Jiang, X.H., Sun, Y., Wang, J.: Color image segmentation: advances and prospects. *Pattern Recogn.* **34**(12), 2259–2281 (2001)

4. Chin-Wei, B., Rajeswari, M.: Multiobjective optimization approaches in image segmentation—the directions and challenges. *Int. J. Advance. Soft Comput. Appl* **2**(1), 40–64 (2010)
5. De, S., Bhattacharyya, S., Chakraborty, S., Dutta, P.: Image segmentation: a review. *Hybrid Soft Computing for Multilevel Image and Data Segmentation*. CIMA, pp. 29–40. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47524-0_2
6. Gower, J.C., Ross, G.J.: Minimum spanning trees and single linkage cluster analysis. *Appl. Stat.* **18**(1), 54–64 (1969)
7. Maji, S., Vishnoi, N.K., Malik, J.: Biased normalized cuts. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2057–2064. IEEE (2011)
8. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Eighth IEEE International Conference on Computer Vision*, vol. 2, pp. 416–423. IEEE (2001)
9. Ooi, W., Lim, C.: Multi-objective image segmentation with an interactive evolutionary computation approach. *J. Intell. Fuzzy Syst.* **24**(2), 239–249 (2013)
10. Rand, W.M.: Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **66**(336), 846–850 (1971)
11. Ripon, K.S.N., Siddique, M.N.H.: Evolutionary multi-objective clustering for overlapping clusters detection. In: *IEEE Congress on Evolutionary Computation*, pp. 976–982. IEEE (2009)
12. Ripon, K.S.N., Tsang, C.H., Kwong, S., Ip, M.K.: Multi-objective evolutionary clustering using variable-length real jumping genes genetic algorithm. In: *18th International Conference on Pattern Recognition*, pp. 1200–1203. IEEE (2006)
13. Shirakawa, S., Nagao, T.: Evolutionary image segmentation based on multiobjective clustering. In: *IEEE Congress on Evolutionary Computation*, pp. 2466–2473. IEEE (2009)
14. Umnikrishnan, R., Pantofaru, C., Hebert, M.: Toward objective evaluation of image segmentation algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(6), 929–944 (2007)
15. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength pareto evolutionary algorithm for multiobjective optimization. In: *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pp. 95–100. International Center for Numerical Methods in Engineering (2001)

Face Recognition by RBF with Wavelet, DCV and Modified LBP Operator Face Representation Methods

J. Jebakumari Beulah Vasanthi¹(✉) and T. Kathirvalavakumar²(✉)

¹ Department of Computer Science, ANJA College, Sivakasi, Tamilnadu, India
jebaarul07@gmail.com

² Department of Computer Science, VHNSN College, Virudhunagar,
Tamilnadu, India
kathirvalavakumar@yahoo.com

Abstract. A face recognition system must be robust with respect to many variability such as viewpoint, illumination, and facial expression of the face image. The main aim of the proposed work is to represent and recognize face images with different poses. An efficient face recognition system with face image representation using wavelet and averaged wavelet packet coefficients in the form of Discriminative Common Vector (DCV) and modified Local Binary Patterns (LBP) and recognition using radial basis function (RBF) neural network is presented. Face images are decomposed by 2-level two-dimensional (2-D) wavelet and wavelet packet transformation. The discriminative common vectors are obtained for wavelet and averaged wavelet packet coefficients. Newly proposed LBP operator is applied on the DCV and LBPs are obtained. Histogram values are generated for the LBP and recognized using RBF network. The proposed work is tested on three standard face databases namely Olivetti-Oracle Research Lab (ORL), Japanese Female Facial Expression (JAFFE) and Essex face database. The extracted features are recognized by the proposed method results in good recognition rates. The execution time for the proposed methods is also less because of the meaningful extracted features obtained from the face representation methods.

Keywords: Face recognition · Wavelet · Discriminative Common Vector
Local Binary Patterns · Radial Basis Function neural network

1 Introduction

This Face recognition is one of the most thrust research areas in the pattern recognition and computer vision. An effective face representation is a key issue for face recognition. The face representation methods based on global features, subspace methods and spatial-frequency techniques are available and used in face recognition system. Subspace-based methods are Principal Component Analysis (PCA), Fisher's Linear Discriminant (FLD) [3] and Independent Component Analysis (ICA) which are widely recognized as the effective and successful face representation methods.

Wavelet domain such as wavelet transform [8] and Gabor wavelet [1] attracted much attention for face recognition. Zhang et al. have proposed face recognition system using wavelet subband representation and kernel associative memory [12]. The compact and meaningful feature vectors are constructed using Wavelet Packet Transformation and is also used in face recognition [6]. Discriminant analysis and Common Vectors have been proposed for face recognition problems. Cevikalp et al. have proposed a face recognition method called the Discriminative Common Vector. The within-class scatter matrix of the samples are used to obtain the discriminative common vectors [4]. Kar et al. have presented a technique [10] by which high-intensity feature vectors extracted from the Gabor wavelet transformation of frontal face images has combined together with Independent Component Analysis (ICA) for enhanced face recognition.

Local binary pattern (LBP) is a nonparametric descriptor, which efficiently summarizes the local structures of images. In recent years, it has aroused increasing interest in many areas of image processing and computer vision and has shown its effectiveness [7]. Local Binary Pattern (LBP) method is applied for human face feature extraction and features are computed from micro patterns created using modified LBP from human faces [3].

Neural based classifiers such as Radial Basis Function neural networks (RBF) are used in pattern recognition and classification processes [5]. Automatic recognition of real time face and mouth in video sequences using RBF has been described by Balasubramanian et al. [2]. A face recognition approach based on kernel discriminative common vectors (KDCV) and RBF network is proposed [9].

Sharma et al. have presented an efficient face recognition method where enhanced local Gabor binary pattern histogram sequence has been used for efficient face feature extraction and generalized neural network with wavelet as activation function is being used for classification [11].

A face recognition system is presented in this paper using RBF with a combination of face representation methods such as wavelet, wavelet packet transformation, DCV and LBP. The paper is organized as follows: the subsequent representation stages are described in Sect. 2. The proposed recognition process using RBF is in Sect. 3. The data set and experiment results along with discussions are presented in Sect. 4.

2 Face Representation

An effective face representation scheme using wavelet, wavelet packet transformation, DCV and newly proposed LBP operator is described in this work. In the first step, Wavelet and Wavelet packet transformation were used on the face images in order to get a dimensionally reduced face features. In the second step of the proposed work, the DCV values are obtained from the wavelet coefficients using the within class scatter matrix method. In the third step, newly proposed LBP operations are done on the DCV coefficients and the resultant LBP values are used to generate the histogram values. The histogram values obtained are given as input to the RBF network for recognition stage.

2.1 Wavelet Transformation

Wavelet transformation is applied for getting through a of filter bank stages. In level-1 of wavelet transformation, the face image is applied to low pass and high pass filter and the values are down sampled by a factor in the horizontal direction. In level-2, the filtered output of level-1 is then filtered by an identical filter pair in the vertical direction. The decomposition of the image into four frequency subbands such as approximation (LL) and detailed coefficients (HL, LH, and HH). The approximation wavelet coefficient is known as low frequency subband and the changes in pose or scale of a face affect only their low-frequency spectrum. The low frequency approximation wavelet coefficients are only considered for further processing in this proposed work.

2.2 Wavelet Packet Transformation

Wavelet packet Transformation is an extension of the wavelet Transformation and results in good recognition performance. The wavelet packet transformation is also used in this proposed work. In wavelet transformation, a face image is decomposed into approximation and detailed coefficients and then the approximation coefficient is only used for the further level decomposition. But in the wavelet packet transformation, both approximation coefficients as well as detailed coefficients of the face are used in the second level for decomposition and so on.

After applying the wavelet packet transformation on the face images for two levels and four wavelet packet coefficients namely approximation, horizontal details, vertical details, and diagonal details are obtained. In the proposed work, the four coefficients are added and the values are averaged.

2.3 Discriminative Common Vector

In this proposed work, in order to obtain a low-dimensional feature representation with enhanced discrimination power, discriminant features are obtained from the wavelet and wavelet packet coefficients using within-class scatter matrix method. Discriminative common vector for each individual class is obtained by removing all the features that are in the direction of the eigenvectors corresponding to the nonzero eigen values of within-class scatter matrix of all classes.

Let the training set be composed of C classes, where each class contains N samples. Let x_m^i denotes m^{th} sample of i^{th} class. Within-class scatter matrix of the samples is constructed to obtain feature vectors, which is defined as [4].

$$S_w = BB^T \quad (1)$$

where the matrix B is given by

$$B = [x_1^1 - \mu_1, \dots, x_N^1 - \mu_1, x_1^2 - \mu_2, \dots, x_N^C - \mu_C] \quad (2)$$

where x_i^j is i -th sample of class j and μ_j is mean of the samples in the j^{th} class.

Let us define $Q = [\alpha_1, \dots, \alpha_r]$, which is the set of orthonormal eigenvectors corresponding to the non-null eigenvalues of S_w and r is the dimension of S_w . Next choose an input sample and project it on the null space of S_w in order to get the common vectors, defined as:

$$x_{com}^i = x_m^i - Q\bar{Q}x_m^i \quad (3)$$

where $m = 1 \dots N$ samples and $i = 1 \dots C$ classes. Calculate the principal components of S_{com} (the eigenvectors w_k , which correspond to the non zero eigenvalues as defined as:

$$J(W_{opt}) = \arg \max_w [W^T S_{com} W] \quad (4)$$

where S_{com} is computed as

$$S_{com} = B_{com} B_{com}^T \quad (5)$$

where B_{com} is given by

$$B_{com} = [x_{com}^1 - \mu_{com} \cdot \dots \cdot x_{com}^C - \mu_{com}] \quad (6)$$

The Feature Vector for training samples is calculated as

$$\Omega_i = W^T x_m^i \quad (7)$$

Similarly, feature vector of test image x_{test} , is obtained by

$$\Omega_{test} = W^T x_{test} \quad (8)$$

The above method is summarized as follows:

1. Compute nonzero eigenvalues and corresponding eigenvectors of S_w using Eqs. (1) and (2).
2. Choose an input sample from each class and project it onto the null space of S_w to obtain the common vectors.
3. Compute x_{com}^i using Eq. (3).
4. Compute the eigenvectors corresponding to the nonzero eigenvalues, by using the Eqs. (4) and (5).
5. Calculate the feature vector for training set and test set using Eqs. (7) and (8) respectively.

2.4 Proposed Local Binary Pattern

The LBP features of face image provide the micro level patterns. The original LBP operator which assigns a label to every pixel except the edge pixels of an image by threshold the 3×3 neighborhood of each pixel with the center pixel value and considering the result as a binary number.

In this proposed work, a modified computation is done for finding the LBP features. A binary bitmap is calculated for each pixel except the edge pixels using 3×3 neighborhood elements. The average value for 3×3 map is calculated and is used as a threshold which is compared with each element of the 3×3 map including the central point. If the averaged value is greater than the averaged value binary map set with 1 value, otherwise 0 is the binary map value. The binary map created with new LBP operator is illustrated in Fig. 1.

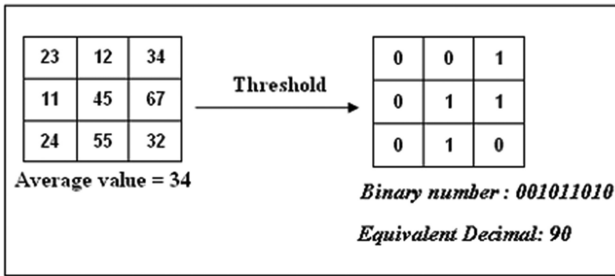


Fig. 1. Proposed LBP operator

A binary pattern is obtained by concatenating all these binary values by a row major starts from left. The corresponding decimal value of the generated binary number is then used for labeling the given pixel. The binary numbers obtained are referred as the LBPs. In Fig. 1, the decimal value for local binary pattern 001011010 is 90. LBPs are obtained by applying the proposed new LBP operator defined in 3×3 neighborhood. The histogram values of the local binary patterns are supplied as input pattern to the RBF for recognition.

In the preprocessing, both wavelet and wavelet packet transformations are performed. The proposed work has two methods. The face images are decomposed by 2-level two-dimensional (2-D) wavelet transformation in the first method. The discriminative common vectors are obtained for wavelet coefficient. Using the new proposed LBP operator the LBPs are obtained and then histogram values are generated. Similarly, in the second method, the wavelet packet coefficients are computed for the face images using 2-level wavelet packet transformation and the resultant coefficients are averaged. DCV and LBP values are computed for the averaged wavelet packet coefficients. The entire proposed work is summarized as follows.

Proposed Method – I (Wavelet + DCV + new LBP + RBF)

1. Perform the steps 2–4 for all the training samples.
2. Apply wavelet transformation and compute the wavelet coefficients.
3. Determine the DCV Coefficients for the wavelet coefficients by the with-in class scatter matrix method.
4. Find the Local Binary Pattern using the new proposed LBP operator.
5. Compute the Histogram values for the LBPs obtained in step 4.
6. Train the RBF network for recognizing the generated values of step 5 using algorithm given in Sect. 3.1.

7. Repeat the steps 2–6 for the test samples.
8. Classify the test image using trained RBF network.

Proposed Method – II (Averaged Wavelet Packet + DCV + new LBP + RBF)

1. Compute the wavelet packet coefficients by applying the wavelet packet transformation.
2. Add the four nodes of A_0^1 namely $A_0^2, D_{0h}^2, D_{0v}^2, D_{0d}^2$ and compute its average.
3. Perform the steps 3–8 of Method – I (Wavelet + DCV + new LBP + RBF).

3 Recognition by Radial Basis Function Neural Network

Radial Basis Function neural network (RBF) contains three layers: input, hidden and output. The number of nodes in the input layer corresponds to the size of input vector.

The input values are supplied to each of the neurons in the hidden layer. The basis function of the hidden layer neurons are considered to be Gaussian and the computed basis function output are passed to the output layer. The hidden layer output is calculated as

$$\varphi_j(X) = \exp\left\{-\frac{\|X - \mu\|^2}{\sigma^2}\right\} \tag{9}$$

where $X = (x_1, x_2, \dots, x_n)^T$ is the normalized input vector, μ is the center and σ is the width. The output of the output layer is computed as

$$y_i = \sum_{j=1}^k w_{ji} \varphi_j(X) \tag{10}$$

where k is the number of hidden neurons, w_{ji} are the weights connecting the hidden layer neuron j and output layer neuron i . The weights are adjusted using the formula,

$$w(t + 1) = w(t) + \lambda(di - yi)\varphi_j(X) \tag{11}$$

where λ is a positive learning rate parameter and d_i is the desired output.

3.1 Algorithm

The training algorithm of Radial Basis Function Network is given as follows.

- Step 1. Generate random number to initialize the weights of the RBF.
- Step 2. Compute hidden layer output using the Eq. (9).
- Step 3. Compute the output layer output using the Eq. (10).
- Step 4. Find the error as the difference between desired and actual output obtained.
- Step 5. Adjust the hidden layer weights according to Eq. (11).

- Step 6. Find output of the output layer.
- Step 7. Compute sum of squared error of the network.
- Step 8. Repeat steps 2–7 for all input patterns.
- Step 9. Repeat steps 2–8 until the acceptable minimum error level is reached.

4 Results and Discussions

The proposed work has been carried out using MATLAB 7.1. The proposed system is tested using face databases such as ORL, The Japanese Female Facial Expression (JAFPE) and Essex Face database.

The ORL face data base contains 40 faces of size 112×92 and each face has 10 different facial views representing various expressions, small occlusion by glasses, different scale and orientations. Hence, there are totally 400 face images in the data-base. In the proposed methods are tested using 5 different poses of 20 person's faces in training stage and 5 different poses of 20 person's faces are used for testing. The Japanese Female Facial Expression contains different facial expressions of Japanese female models. For training, 5 different poses of 15 models are used and the other 5 different poses of 15 models are used for testing. The dimension of the face image is 256×256 . The Essex Face database is containing faces of more than 150 male and female with 20 images per individual of <http://cswww.essex.ac.uk/mv/allfaces/index.html> University of Essex, UK with the size of 180×200 . The 5 different poses of 20 person's face are used for the training and another 5 different poses of 20 person's face images are used as input for the testing. The 5 poses for training and 5 poses for testing are sequentially considered.

The original image, LBP image and the resultant image after applying wavelet transformation and new proposed LBP operator with its histogram are displayed in Fig. 2.



Fig. 2. Original image, LBP image, Wavelet + new LBP image and Histogram of the new LBP image

The recognition rates are obtained by doing the training and then testing process repeatedly for 25 times and averaged. The recognition rates of the three databases for different wavelets namely Haar, Sym4, Sym8, Db4, Db6, Coif2, Coif4 are shown in

Table 1. For the proposed method – I, when Haar wavelet is used, the highest recognition rate of 98% is obtained for all the three databases but lowest is 94% for Db6 in both ORL and JAFFE database and 95.4% for Sym8 in Essex database. When comparing the recognition rates of the two methods, the recognition rate for haar wavelet is 98.3% in method – II which is greater than the proposed method – I.

Table 1. Recognition rates of proposed Method – I and Method – II

Wavelet name	Recognition rate of Method – I			Recognition rate of Method – II		
	ORL database	JAFFE database	Essex database	ORL database	JAFFE database	Essex database
Haar	98.0	98.0	98.0	98.3	98.0	97.0
Sym4	97.0	97.33	96.3	97.0	96.7	96.0
Sym8	96.33	96.66	95.4	96.66	96.3	94.3
Db4	96.66	96.0	96.3	97.0	96.0	96.0
Db6	94.0	94.0	96.0	97.0	95.0	97.0
Coif2	95.67	96.0	97.0	96.3	96.6	97.3
Coif4	96.0	97.0	96.0	95.0	94.66	95.6

The computed preprocessing time and training time are shown in Table 2 for both the proposed methods. The preprocessing time is more than the training time because of the computation involved. The preprocessing time for JAFFE database is more due to the high dimension. Though the preprocessing time is different for different databases, the training time is less and not having much difference between the three databases. Preprocessing time and training time for the second method are shown in Table 2. When comparing the preprocessing time, the first method's time is less than the second method. Training time for Essex database of the proposed method – II consumes less time.

Table 2. Pre processing and training time of proposed methods

Wavelet name	Method – I						Method – II					
	Pre-processing time in seconds			Training time in seconds			Pre processing time in seconds			Training time in seconds		
	ORL	JAFFE	Essex	ORL	JAFFE	Essex	ORL	JAFFE	Essex	ORL	JAFFE	Essex
Haar	4.281	13.664	7.174	1.147	1.301	1.414	7.543	24.308	16.093	1.142	1.718	1.368
Sym4	5.202	15.519	8.686	1.241	1.554	1.491	8.757	28.931	16.878	1.253	1.855	1.382
Sym8	5.872	21.697	10.854	1.280	1.946	1.656	9.492	37.114	21.653	1.322	1.988	1.569
Db4	5.011	15.576	8.668	1.220	1.669	1.500	8.821	28.322	16.112	1.265	1.879	1.484
Db6	5.374	18.158	9.756	1.257	1.938	1.610	8.608	33.072	19.277	1.190	1.943	1.533
Coif2	5.388	17.888	10.050	1.287	1.859	1.623	8.717	32.899	19.633	1.201	1.912	1.519
Coif4	6.913	28.091	14.481	1.353	2.120	1.736	11.46	49.62	27.050	1.510	2.245	1.649

The recognition rates obtained for the ORL, JAFFE and ESSEX databases on applying other methods are compared and displayed in Table 3. The recognition rate 97.54% is obtained for Push-Pull Marginal Discriminant Analysis method on ORL database. The MLA + NM method resulted with recognition rate of 97% on JAFFE database. Recognition rate of 97.2% has been achieved by Curvelet with SVM method for ESSEX database. The methods such as Eigen face, wavelet face and SOM have improved results when combined with neural networks (NN) or convolution neural networks.

Table 3. Recognition rates of other methods vs proposed method

Method name	ORL	Method name	JAFFE	Method name	Essex
Eigen faces	89.5%	MLA + NN	91.14	Wavelet + HMM	84.2%
Direct LDA	90.8%	LDA + SVM	91.27%	DWT + PCA	86.1%
Eigen faces + NN	91.2%	SVM	91.6%	PZM	88.02%
Waveletface + NN	93.5%	Adaboost	92.4%	Gabor + SHMM	88.7%
Gabor + rank correl.	96%	LBP + SVM	92.0	DM	91.72%
2DPCA	96%	PCA + SVM	93.43%	Fisher faces	92.62%
Push-Pull Marginal Discriminant Analysis	97.54%	MLA + NM	97%	Curvelet + SVM	97.2%
Proposed Method – I	98.0%	Proposed Method – I	98.0%	Proposed Method – I	98.0%
Proposed Method – II	98.3%	Proposed Method – II	98.0%	Proposed Method – II	97.3%

The recognition rates obtained for proposed method – I and method – II for the ORL, JAFFE and Essex face databases for different wavelets are shown in Fig. 3.

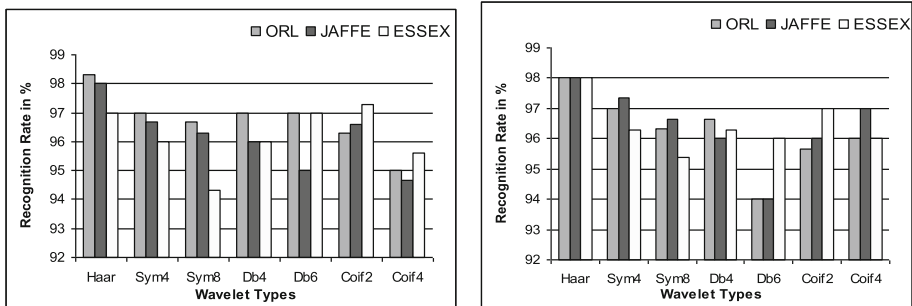


Fig. 3. Comparison of recognition rates – Method I and Method II

In the proposed system, dimensionality reduction of the face image has been achieved by wavelet based transformations. Using DCV method, Discriminant values are obtained from the dimensionality reduced image. Micro patterns of the face image

are obtained using newly proposed LBP operator. Finally, RBF network recognized the histogram which is obtained from the micro patterns. This good recognition rate is because of the extracted discriminant features from the efficient face representation methods in the proposed work.

5 Conclusion

In face recognition system, constructing an effective face representation is a key issue as it plays an important role in face recognition. A face recognition system for different poses is devised with effective and collective face representation methods. The face representation includes the approach of wavelet or wavelet packet transformation, discriminative common vectors and newly proposed LBP operator. The generated histogram values from the LBP features are recognized by RBF network. The recognition rate is not affected by the combined method used in the face representation stage. Since the number of features has been reduced the computational complexity is also minimized. The proposed work yielded the better recognition rates for ORL, JAFFE and ESSEX face databases. The improved recognition rate is due to the combined face representation methods which provide necessary discriminate information for classification and RBF network.

References

1. Abdulrahman, M., Gwadabe, T.R., Abdu, F.J., Eleyan, A.: Gabor wavelet transform based facial expression recognition using PCA and LBP. In: Signal Processing and Communications Applications Conference (SIU), pp. 2265–2268 (2014)
2. Balasubramanian, M., Palanivel, S., Ramalingam, V.: Real time face and mouth recognition using radial basis function neural networks. *Expert Syst. Appl.* **36**, 6879–6888 (2009)
3. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs fisher faces: recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(7), 711–720 (1997)
4. Cevikalp, H., Neamtu, M., Wilkes, M., Barkana, A.: Discriminative common vectors for face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(1), 4–13 (2005)
5. Er, M.J., Wu, S., Lu, J., Toh, H.L.: Face recognition with radial basis function (RBF) neural networks. *IEEE Trans. Neural Netw.* **13**(3), 697–710 (2002)
6. Garcia, C., Zikos, G., Tziritas, G.: Wavelet packet analysis for face recognition. *Image Vis. Comput.* **18**, 289–297 (2000)
7. Huang, D., Shan, C., Ardabilian, M., Wang, Y., Chen, L.: Local binary patterns and its application to facial image analysis: a survey. *IEEE Trans. Syst. Man Cybern.* **41**(6), 765–781 (2011)
8. Huang, Z.H., Li, W.J., Wang, J., Zhang, T.: Face recognition based on pixel-level and feature-level fusion of the top-level's wavelet sub-bands. *Inf. Fusion* **22**, 95–104 (2015)
9. Jing, X., Yao, Y., Yang, J., Zhang, D.: A novel face recognition approach based on kernel discriminative common vectors (KDCV) feature extraction and RBF neural network. *Neurocomputing* **71**, 3044–3048 (2008)

10. Kar, A., Bhattacharjee, D., Basu, D.K., Nasipuri, M., Kundu, M.: High performance human face recognition using independent high intensity Gabor wavelet responses: a statistical approach. *Int. J. Comput. Sci. Emerg. Technol.* **2**(1), 178–187 (2011)
11. Sharma, P., Arya, K.V., Yadav, R.N.: Efficient face recognition using wavelet-based generalized neural network. *Image Vis. Comput.* **28**(1), 177–187 (2010)
12. Zhang, B., Zhang, H., Ge, S.S.: Face recognition by applying wavelet subband representation and kernel associative memory. *IEEE Trans. Neural Netw.* **15**(1), 166–177 (2005)

DNN-HMM Acoustic Modeling for Large Vocabulary Telugu Speech Recognition

Vishnu Vidyadhara Raju Vegesna^(✉), Krishna Gurugubelli,
Hari Krishna Vydana, Bhargav Pulugandla, Manish Shrivastava,
and Anil Kumar Vuppala

Speech Processing Lab, LTRC, International Institute of Information Technology,
Hyderabad, Hyderabad, India

{vishnu.raju,krishna.gurugubelli,hari.vydana,
bhargav.pulugandla}@research.iiit.ac.in,
{m.shrivastava,anil.vuppala}@iiit.ac.in

Abstract. The main focus of this paper is towards the development of a large vocabulary Telugu speech database. Telugu is a low resource language where there exists no standardized database for building the speech recognition system (ASR). The database consists of neutral speech samples collected from 100 speakers for building the Telugu ASR system and it was named as IIIT-H Telugu speech corpus. The speech and text corpus design and the procedure followed for the collection of the database have been discussed in detail. The preliminary ASR system results for the models built in this database are reported. The architectural choices of deep neural networks (DNNs) play a crucial role in improving the performance of ASR systems. ASR trained with hybrid DNNs (DNN-HMM) with more hidden layers have shown better performance over the conventional GMMs (GMM-HMM). Kaldi tool kit is used for building the acoustic models required for the ASR system.

Keywords: DNNs · HMMs · GMM · ASR · MFCCs

1 Introduction

The presence of high variability in human speech makes speech recognition as a challenging task. This variability is due to different attributes of the speaker (i.e. gender, emotions, health status), various accents and uncertainty in training and testing environments. In ASR, the indefinite length speech signals are mapped as sequence of phonetic symbols or words. HMM models were very efficient in dealing the indefinite length sequences and are extensively used to model the sequential patterns. This modeling is done by considering the particular state sequences associated with the observations at a particular probability distribution. The estimation of these probability distribution which are associated with HMM states is done using GMMs. The development of ASR systems for the real world tasks was made possible with Expectation maximization (EM) algorithm

along with the generative methods such as GMM-HMM. GMMs represent the relation between the acoustic input and states of HMM. [17] made an attempt in which GMMs were constrained to improve the assessment time. To resolve the overfitting issue, the trade off is maintained among the assessment speed and the training data size. Improvement in the accuracy of the GMM-HMM system is observed when the input features i.e. Mel frequency cepstral coefficients (MFCCs) are concatenated with the bottle neck or tandem features [9]. The inefficiency of the GMMs lies in modeling the data when they lie on or near the nonlinear manifold. Apart from generative training methods (GMM-HMM), several discriminative training methods with the EM algorithm [8, 11] were employed which yielded better results in word error rate (WER) for HMM based ASR systems.

Artificial neural networks (ANN) has the potential to learn and build models for overcoming the inefficiency of GMMs. Training of these ANNs is done by backpropagating the error derivatives. The HMM states were predicted from the several frame coefficients representing the acoustic input. This prediction was done using ANNs by considering a single layer of non-hidden units [3]. In the recent days, HMM models that use ANNs instead of the existing GMMs have gained more prominence [4, 5, 13]. The usage of single hidden layer in ANNs was not sufficient enough to challenge GMMs. The ANNs were limited to provide the bottle neck features for building the ASR systems [2, 6, 7].

Deep neural networks (DNNs) are trained with many non-linear hidden units and a large output layer [10]. A large number of HMM states are accommodated at the output layer. The reason for accommodating more number of HMM states is due to the modeling of triphone HMMs where the phones on either side are taken into account. The different states of these triphone HMMs are combined or tied together to generate many number of tied states. In this paper, a feed forward DNN models trained with cross entropy [1] are considered for the acoustic modelling of the large vocabulary Telugu speech ASR system. The speech samples are taken from large vocabulary Telugu speech corpus for the evaluation of ASR system. The acoustically trained DNN-HMM ASR system performance is compared with the existing GMM-HMM trained system.

Remaining parts of the paper are described as follows: Details of Telugu speech corpus is discussed Sect. 2. In Sect. 3 overview of the proposed system is given. Section 4 describes the DNN-HMM acoustic modelling. In Sect. 5, proposed ASR system is evaluated and presented. Conclusion and future scope are discussed in Sect. 6.

2 IIIT-H Telugu Speech Corpus

In this paper, the major focus is to collect speech samples for the low resource Telugu language. There is no standard database for building the Telugu ASR system. This IIIT-H corpus of neutral emotional speech has been collected with a motto to meet the needs of proper Telugu ASR system building. This neutral speech corpus and the text corpus design have resulted from the joint efforts of Speech Processing Lab at IIIT-Hyderabad, India.

In order to build the large vocabulary Telugu speech corpus, neutral speech samples are collected from speakers of different regions to have taken care of different dialects. The Telugu language is widely spoken in the states of Telangana and Andhra Pradesh. The speech samples are collected from the speakers of different regions in these two states. The speakers from 8 different dialect regions are considered in this corpus. The speech corpus consists the speech from 100 speakers which comprises of 60 male and 40 female speakers. Among these 100 speakers 34 speakers speech samples were collected in a noise-free recording studio with help of a Zoom voice recorder. The sampling rate of all the speech samples is maintained at 16,000 samples/second. Another 56 speakers speech samples are collected from DD-Telugu news corpus and the remaining 10 speakers neutral speech samples are considered from emotional IITKGP-SESC Telugu corpus. The total utterances from speech samples considered by combining all the speakers are around 14,560. The total duration of speech samples collected is around 10 h. A total of 25,700 unique words were considered in the speech corpus.

In the process of collecting the speech samples from the first set of 34 speakers, they have been provided with the text material in the IIT-H prompts a day before the recordings. These 34 speakers were students from the institute of IIT-Hyderabad. The age group of these speakers was from 19–30 years. The text material consists of 500 text sentences which differ from one speaker to the other. Each speaker was given a choice to select his own 150 text sentences from the given 500 text sentences and those 150 text sentences were spoken by the speaker in the recording studio. A separate set of 20 text sentences along with the chosen 150 sentences is spoken in common by every speaker.

The text material in these IIT-H prompts consists of phonetically compact sentences which were designed in such a way to provide a proper and entire coverage of the pairs of phones in the Telugu language. A total of 5,780 utterances are recorded from these speakers. In the second set of 56 speakers, speech samples were collected from DD-Telugu news corpus. The reason for considering the speech samples from DD-news corpus is to balance the IIT-H Telugu corpus with experienced artists alongside the student’s data. The age group of these 56 speakers was from 32–60 years. The utterances considered from this set of speakers are around 7280. The other set of samples are considered from IITKGP-SESC emotion corpus. The neutral speech samples from the 10 speakers are considered, where each speaker has spoken 15 sentences. The data were collected in 10 sessions from each speaker. A total of 1500 utterances were considered and added to the existing speech data. The main reason for considering the speech samples from other databases is to increase the size from small vocabulary to a large vocabulary. Speakers from the age group of 19–60 years were considered from both the two Telugu speaking states covering the 8 different dialects. All the speech files are recorded in ‘wav’ format. The associated orthographic transcription of the words spoken by the speaker are provided with ‘.txt’ for the corresponding ‘wav’ files.

3 System Overview

The overview of the proposed system is clearly shown in Fig. 1. This system includes DNN-HMM acoustic modeling. To perform this model training, the processing of speech samples are done to extract the MFCC and log Mel-filterbank (FilterBank) features. For directly modeling the DNN should be trained with input FilterBank features and the HMM states are the learning target. The final decision about the speech in the testing phase is taken by considering the average output of the neural networks for each input frame features of the speech.

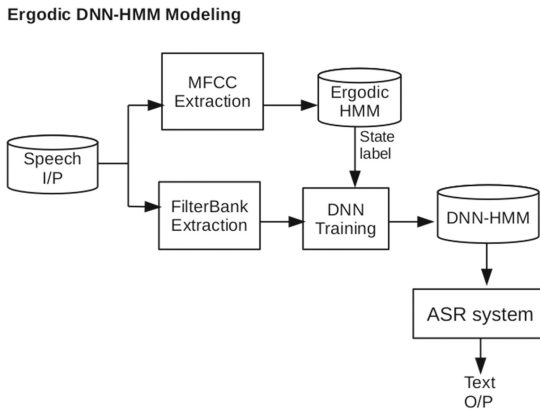


Fig. 1. System overview

The acoustic label with the best result is chosen. In this paper, the concentration is towards ergodic DNN-HMM modeling [1] where the ergodic HMMs are trained in the first stage using MFCC features per acoustic class and perform the classification using the maximum likelihood classification [16]. In the next stage, the state labels are produced from HMM models by performing forced alignments [18]. In building the hybrid DNN-HMM system the FilterBank features are the input and the HMM states are considered to be the learning targets. The DNN-HMM based experiments are implemented using Kaldi toolkit [14].

4 DNN-HMM Modeling

In this section the implementation of DNN and ergodic HMM are explained in detail. The training and testing stages of the DNN-HMM are also explained.

4.1 Deep Neural Networks

DNNs are popular form of feed forward ANNs with more than one hidden layer. These networks are mostly employed for acoustic classification tasks shown in [1]. For the objective to classify a feature of interest y among the available Q classes, DNNs are helpful in estimating the probabilities p_i , where $i \in \{1 \dots Q\}$ of each class for the given input y . The input features are the time-frequency representation of the input signal, such as MFCCs and Mel-filterbanks. The consecutive feature frames are concatenated through the sliding window to provide the acoustic context to the DNN. Figure 2 shows the graphical representation of the DNN architecture for classification. For a hidden layer (H-layer) DNN, the non-linear function is computed as below,

$$g_W(y) = g_H(W_H g_H(W_{h-1} \dots g_1(W_1 y))), \quad (1)$$

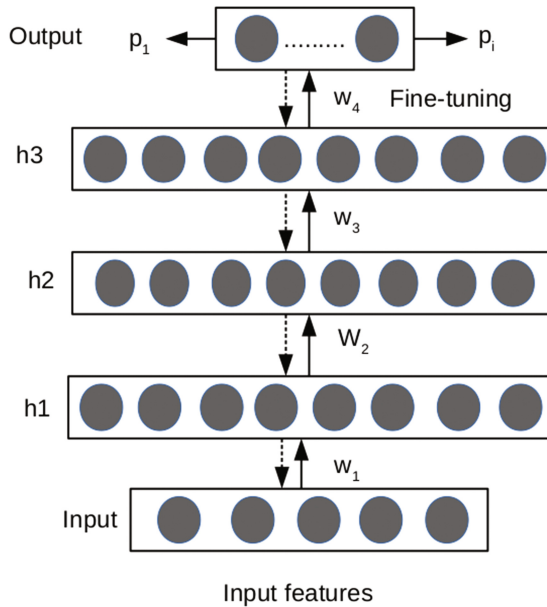


Fig. 2. A DNN architecture described by an input, a given number of hidden layers and the output probabilities

where y are the input features, g_h , $h = 1, \dots, H$ represent the activation function and W_h , $h = 1, \dots, H$ are the corresponding DNN weights. Indexes with $h = 1, \dots, H - 1$ represent the number of hidden layers. Sigmoidal functions are used as the hidden layer activations. Index H layer represents the output layer with activations g_H . The softmax activation function is used in the computation

of output for the classification task. The training of DNN weights is done by minimizing the cost function of cross entropy which is given as below,

$$C = - \sum_{i=1}^Q q_i \log p_i \quad (2)$$

where C is the Cross entropy cost function, p_i denotes the output of softmax and q_i is the target.

4.2 Ergodic HMMs

HMMs are the effective parametric representation for the given time series observation. In ergodic HMMs the sequential property of the observations are modeled. The uncertainty in the structure of the speech [15] has made these ergodic HMMs more suitable for the classification. The classification is done by considering the training of GMM-HMM for all classes with features as MFCC coefficients. These GMM-HMMs with a set of states represent each class. The parameters of GMM-HMMs for all the classes are learnt according to maximum likelihood estimation. Baum-Welch and EM algorithms are adopted to estimate the state prior and transition probabilities along with the weight/mean/covariance parameters of GMMs. For decoding the HMM state sequences, viterbi algorithm is used.

4.3 Training for DNN-HMM

In recent years, hybrid DNN-HMM systems [1, 19] have been adopted for the speech recognition task. These hybrid systems have a advantage of DNNs strong learning power and HMMs sequential modeling ability to outperform the existing GMM-HMM systems. The DNN-HMM system require state labels for the subsequent training of DNN models. The steps for the parameter learning using the cross entropy criteria is as below,

Step1: Initialization. The set of GMM-HMMs for all the classes are learnt using maximum likelihood estimation criterion. The HMM topology is generated based on GMM-HMM model, which is used for the hybrid DNN-HMM system which include state prior and transition probabilities.

Step2: Forced Alignment. The main reason for performing the forced alignment is generate the frame-level state labels by matching the acoustic speech with the corresponding labels from the viterbi algorithm with GMM-HMM. The state labels are the learning targets of DNN output layer.

Step3: Cross Entropy Training Criterion. The DNNs are trained in such a way that the estimation of the posterior probabilities of the HMM states from the given observation which is shown in Fig. 3. Cross entropy training is done by applying the criterion which is discussed in Sect. 4.1.

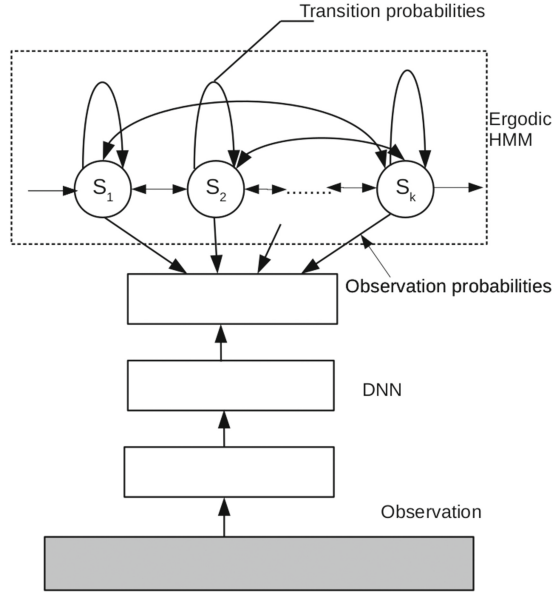


Fig. 3. Architecture of DNN-HMM hybrid system.

4.4 Testing for DNN-HMM

Posterior probabilities of states at the given observations are generated after decoding the DNN. During the decoding process, HMMs requires the likelihood estimation instead of the generated posterior probabilities. The posterior probabilities are converted to likelihood estimates. The HMM state sequences are decoded by using the viterbi algorithm [12].

5 Results and Discussion

The speech recognition performance of hybrid DNN-HMM system is examined on the large vocabulary Telugu speech Corpus. The dataset consists of 10 h of speech duration. In the first step, a conventional triphone based HMM model is built with 39-dimensional MFCCs using 23rd order filter-bank analysis. During the training of DNNs, the state labels of HMMs are considered to be the learning targets. These systems are built using Kaldi tool kit. The learning rate needed to run the first 20 epochs is 0.015 and the reduced learning rate for the next 10 additional epochs is around 0.002. The system training for the DNN-HMM is done through cross entropy criterion discussed in Sect. 4.3. The size of the hiddenlayers is around 300 units for each layer. DNNs having both two and three hidden layers were considered for the evaluation. SRILM toolkit is used to perform language modelling. Trigram language model is considered in the evaluation of an ASR system.

Table 1. WER comparison of GMM-HMM system with Hybrid DNN-HMM system.

Performance of the baseline and hybrid models in WER (%)				
Baseline (GMM-HMM)		Hybrid (DNN-HMM)		
Triphone (LDA+MLLT)	Triphone (LDA+MLLT+SAT)	1-hidden layer (MLP)	2-hidden layers	3-hidden layers
26.52	26.24	23.35	22.20	22.56

Table 1 shows that the hybrid DNN-HMM improves the WER performance over the existing GMM-HMM. Baseline GMM-HMM system WER is reported for the cases of monophone, triphone (LDA+MLLT), triphone (LDA+MLLT+SAT) in C:1–2. The WER for the DNN-HMM system is reported in C:3–5 for one, two and three hidden layers respectively. From the results shown in Table 1 it is observed the best performance i.e. WER is minimum for the case of DNN with two hidden layers shown in C:4. Hence it is observed that the hybrid DNN-HMMs perform better than the conventional GMM-HMM system.

6 Conclusion and Future Scope

In this paper, the advancements in the recent literature has been explored regarding the replacements of the existing GMM-HMM based ASR with DNNs. The DNN structures could directly accommodate some type of speech variability. The hybrid DNN-HMM systems have the advantage of adopting the strong learning power from DNNs and the sequential modeling ability from HMMs. Good results have been achieved in the ASR performance using DNN-HMMs when compared to existing GMM-HMMs. This study can be extended to End-to-End DNN systems replacing the existing HMMs.

Acknowledgements. The authors would like to thank TCS for partially funding the first author for his PhD programme.

References

1. Bao, X., Gao, T., Du, J., Dai, L.R.: An investigation of high-resolution modeling units of deep neural networks for acoustic scene classification. In: Proceedings of International Joint Conference on Neural Networks (IJCNN), pp. 3028–3035. IEEE (2017)
2. Bao, Y., Jiang, H., Dai, L., Liu, C.: Incoherent training of deep neural networks to de-correlate bottleneck features for speech recognition. In: Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6980–6984. IEEE (2013)
3. Bourlard, H.A., Morgan, N.: Connectionist Speech Recognition: A Hybrid Approach, vol. 247. Springer, New York (2012). <https://doi.org/10.1007/978-1-4615-3210-1>

4. Dahl, G., Mohamed, A.R., Hinton, G.E., et al.: Phone recognition with the mean-covariance restricted Boltzmann machine. In: *Advances in Neural Information Processing Systems*, pp. 469–477 (2010)
5. Dahl, G.E., Yu, D., Deng, L., Acero, A.: Large vocabulary continuous speech recognition with context-dependent DBN-HMMS. In: *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4688–4691. IEEE (2011)
6. Deng, L., Seltzer, M.L., Yu, D., Acero, A., Mohamed, A.R., Hinton, G.: Binary coding of speech spectrograms using a deep auto-encoder. In: *Proceedings of Eleventh Annual Conference of the International Speech Communication Association* (2010)
7. Grézl, F., Karafiát, M., Kontár, S., Cernocký, J.: Probabilistic and bottle-neck features for LVCSR of meetings. In: *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, p. IV-757. IEEE (2007)
8. He, X., Deng, L., Chou, W.: Discriminative learning in sequential pattern recognition. *IEEE Sig. Process. Mag.* **25**(5), 14–36 (2008)
9. Hermansky, H., Ellis, D.P., Sharma, S.: Tandem connectionist feature extraction for conventional HMM systems. In: *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, pp. 1635–1638. IEEE (2000)
10. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Sig. Process. Mag.* **29**(6), 82–97 (2012)
11. Jiang, H.: Discriminative training of HMMs for automatic speech recognition: a survey. *Comput. Speech Lang.* **24**(4), 589–608 (2010)
12. Lou, H.L.: Implementing the Viterbi algorithm. *IEEE Sig. Process. Mag.* **12**(5), 42–52 (1995)
13. Mohamed, A.R., Sainath, T.N., Dahl, G., Ramabhadran, B., Hinton, G.E., Picheny, M.A.: Deep belief networks using discriminative features for phone recognition. In: *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5060–5063. IEEE (2011)
14. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al.: The Kaldi speech recognition toolkit. In: *Workshop on Automatic Speech Recognition and Understanding*, No. EPFL-CONF-192584. IEEE Signal Processing Society (2011)
15. Ramasubramanian, V., Karthik, R., Thiyagarajan, S., Cherla, S.: Continuous audio analytics by HMM and Viterbi decoding. In: *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2396–2399. IEEE (2011)
16. Saul, L.K., Rahim, M.G.: Maximum likelihood and minimum classification error factor analysis for automatic speech recognition. *IEEE Trans. Speech Audio Process.* **8**(2), 115–125 (2000)
17. Steve, Y.: Large vocabulary continuous speech recognition: a review. *IEEE Sig. Process. Mag.* **21**, 786–797 (1996)
18. Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., et al.: *The HTK Book*, vol. 3, p. 175. Cambridge University Engineering Department, Cambridge (2002)
19. Yu, D., Deng, L.: *Automatic Speech Recognition: A Deep Learning Approach*. Springer, London (2014). <https://doi.org/10.1007/978-1-4471-5779-3>

Memetic Algorithm Based on Global-Best Harmony Search and Hill Climbing for Part of Speech Tagging

Luz Marina Sierra Martínez^(✉), Carlos Alberto Cobos, and Juan Carlos Corrales

University of Cauca, Popayan, Colombia
{lsierra, ccobos, jcorral}@unicauca.edu.co

Abstract. The task of assigning tags to the words of a sentence has many applications today in natural language processing (NLP) and therefore requires a fast and accurate algorithm. This paper presents a Part-of-Speech Tagger based on Global-Best Harmony Search (GBHS) which includes local optimization (based on the Hill Climbing algorithm that includes knowledge of the problem to define the neighborhood) for the best harmony after each improvisation (iteration). In the proposed algorithm, a candidate solution (harmony) is represented as a vector of the size of the numbers of word in a sentence, while the fitness function considers the cumulative probability of tagging each word and its relation to its predecessor and successor word. The proposed algorithm obtained 95.2% precision values and improved on the results obtained by other taggers. The experimental results were analyzed with Friedman non-parametric statistical tests, with a level of significance of 90%. The proposed Part-of-Speech Tagger algorithm was found to perform with quality and efficiency in the tagging problem, in contrast to the comparison algorithms. The Brown corpus divided into 5 folders was used to conduct the experiments, thereby allowing application of cross-validation.

Keywords: Global-Best Harmony Search · Part-of-Speech Tagging
Hill Climbing · Metaheuristic algorithms · Memetic algorithm

1 Introduction

The tagging problem has become an essential part of natural language processing, given the various applications that nowadays use it as a first step in pre-processing data. Part-of-Speech Tagging (POST) consists of assigning a tag to each word, taking into account its corresponding morphosyntactic category, for example: name, verb, adjective, adverb, article, and determinant, among others [1]. The tagging problem is complex given the contextual limitations of each language [2], for example the ambiguities of words, among other aspects. POST proposals can be classified into statistical methods [3–7], rule-based methods [8, 9], neural network based methods [10, 11], and metaheuristics methods [12–14]. In general terms, the most recognized and best-performing POST are those based on statistical methods. However, the new proposals from the metaheuristic approach have performed very well and seek to be simpler, more efficient, and more robust. As their starting point, these proposals can take rules, statistical information, or both approaches.

This paper presents a memetic algorithm for the POST problem. This algorithm is based on Global-Best Harmony Search as a global optimization strategy, Hill Climbing that does local optimization, and a local neighborhood built with knowledge of the problem, which allows the algorithm to obtain better results than a state-of-the-art algorithm called HSTAGger and a baseline established with an algorithm that performs a random walk.

The rest of the paper is organized as follows: Sect. 2 presents a brief description of related works for the building of POST, tagset and the tagged corpus most used to evaluate English taggers. Section 3 provides a background on the Harmony Search algorithm. The proposed algorithm is then presented in Sect. 4. Section 5 presents the results; and finally, Sect. 6 presents conclusions and intentions for future work.

2 Related Works

2.1 Traditional Approaches to Build POS Taggers

Some work related to POST based on statistical information includes proposals for traditional languages such as English, and non-traditional languages such as Kazakh, Nepali, Tamil, Odia, and Bengali, among others. The most relevant are those that use: (1) Hidden Markov models [3, 4]; (2) models of Maximum Entropy (ME) [4–6, 15]; (3) Trigrams (TnT) [7]; (4) Support Vector Machines [16, 17]; and (5) Conditional Random Fields [18]. This type of tagger has the advantage that it does not require much knowledge about the language for which it is developed and has as its main difficulty complexity both in its construction process and in the computational requirements given the amount of information that must be handled [19].

At the level of rule-based taggers, the Brill proposal is mainly found for English [8, 9], which is the basis for other approaches and proposals. Some relevant works include: specialized POST in source code comments [20], and taggers for Malayalam [21], Hindi [22], and English [23]. These types of POST have very good levels of precision, but their construction is complex since advanced knowledge of the language is required for rule definition.

At the level of POST based on neural network are those proposed by Schmid [10] in 1994 and Nakamura [11] in 1989, which are the basis for other research such as those using artificial neural networks with parameter management for several languages [24, 25] and a tagger for German [26]. These types of proposals have a high dependence on both the data and the knowledge of the language for the refinement of the parameters.

Also, other approaches are found that use combinations of recurrent neural networks and graphs, or modifications that have as starting point POST based on statistical such as the proposal of Zenaki et al. [27], which includes the use of a neural network and tag projection between languages, and its starting point is a TnT tagger [7]; the proposal of Duong et al. [28] that uses a POST based on maximum entropy and parallel alignment of words to obtain a multilanguage approach, which presents mismatches among the tags given the noise that may exist in the parallel alignment. Despite obtaining good results, these are complex proposals.

2.2 Metaheuristic Algorithms as an Approach to Build POS Taggers

Recently, the use of metaheuristic algorithms for developing POST proposals from both statistical and rule-based approaches has taken off. These include: Forsati et al. in 2010, who presented HSTAGger [12], a tagger based on Harmony Search with good results compared to previous taggers; in 2012 BEETAGger was presented [29], a tagger based on Bee Colony Optimization (BCO); in 2015 version II of their tagger HSTAGger was presented [28], modifying some parameters and the fitness function, achieving a slight improvement in the precision of the algorithm. Silva et al. in 2014 presented a tagger based on the evolution of rules using Particle Swarm Optimization (PSO) [30], obtaining improvements in precision compared to their previous proposals. A tagger based on a genetic algorithm, GA-Tagger, was developed in 2012 [14] and also in the same year, a tagger that uses two evolutionary algorithms, one to discover the rules and another to do the tagging, using statistical data in the objective function [31], and in 2013, a POST algorithm based on genetic and PSO algorithms [32] and a tagger based only on PSO [33]. Bachir et al. in 2014 presented a POST based on genetic algorithms for Arabic [34]. Ekbal and Saha in 2013 proposed a tagger based on mono- and multi-objective optimization techniques for Bengali and Hindi [35] and in 2011 presented a multi-objective classifier for Bengali, Hindi, and Telugu [36].

2.3 Tagsets for Tagging Corpus and Tagged Corpus

The set of possible labels for a word for each language may vary according to the contexts and morphological structure of each language, for example: In 2014, Dinakaramani et al., [37] established a set of 23 POS tags to label 10,000 sentences from the IDENTIC corpora of the Indonesian language. In 2012, Petrov et al. [38], proposed the Universal tagset, defining 12 possible tags: NOUN (nouns), VERB (verbs), ADJ (adjectives), ADV (adverbs), PRON (pronouns), DET (determiners and articles), ADP (prepositions and postpositions), NUM (numerals), CONJ (conjunctions), PRT (particles), ‘.’ (punctuation), and X (a catch-all for other categories such as abbreviations or foreign words). And in 2008, Rabbi, et al., [39] presented the procedure followed for the design of a tagset for the Pashto language, obtaining 215 tags.

The following is a brief review of some of the most commonly used English tagged corpus in the literature to evaluate POST: In 1979, Francis and Kucera [40] proposed the Brown corpus for American English, containing 1,014,312 words in text categories. This corpus has a total of 473 categories arising from the subdivisions of the 82 main tags and is widely used for tagging in English. In 1993, Marcus et al. [41] presented the Penn Treebank corpus with a reduction in the tagset in comparison with the tagset of the Brown corpus (48 tags). In 2005, Kohen [42] presented the Europarl corpus extracted from the Proceedings of the European Parliament, which includes versions in most European languages.

3 Harmony Search Algorithm

Evolutionary metaheuristic algorithms use a population of individuals who represent solutions to a complex optimization problem. The population undergoes certain transformations and then a selection process that favors the best to generate a new generation, so that after a specific number of generations, the best individual in the population approaches the optimal solution or reach [43].

Harmony Search (HS) algorithm is an evolutionary metaheuristic that mimics the improvisational process used by musicians to seek the best harmonies on their instruments [44]. It was proposed in 2005 by Lee and Geem [45]. HS is oriented by the Harmony Memory Considering Rate (HMCR) and Pitch Adjusting Rate (PAR) parameters that control the global and local search, an arbitrary distance bandwidth (bw) for changing or mutating some values in new solutions (harmonies), Harmony Memory Size (HMS) that defines the size of the harmony memory (HM) (population) and maximum number of improvisations (NI) that is used as a stopping criterion. It consists of the following steps [45]: (1) Initialize the optimization problem and algorithm parameters, either maximization or minimization, the fitness function, the range of variables, the size of the harmony memory, among others; (2) Initialize the harmony memory (HM) that is usually random; (3) Improvise a new harmony, taking into account three rules: Considering memory, Tone adjustment and Random selection; (4) Update the HM: the new vector generated (improvised) replaces the worst harmony in HM only if its fitness is better. (5) Repeat steps 3 and 4 until the termination criterion is satisfied, usually when the number of improvisations is reached.

In 2007 Mahdavi et al. [46] proposed the improved harmony search algorithm (IHS), which proposes two equations to dynamically update the PAR parameter and bw, which have a significant effect on HS performance. In 2008, Omran et al. proposed the Global-Best Harmony Search (GBHS) algorithm [47], which hybridizes HS with the swarm intelligence concept proposed in PSO, where a swarm of individuals (called particles) flies around the best solution known on search space. GBHS modifies the pitch

```

01 Initialize the problem (Maximization or Minimization) and HS Parameters: HMS, HMCR,
    PAR and NI
02 Initialize HM
03 repeat /* improvise a new harmony */
04   for each  $i \in [1, N]$  do
05     if  $U(0,1) < HMCR$  then /* memory consideration */
06        $x'_i = x^j_i$ , where  $j \sim U(1, \dots, HMS)$ 
07     if  $U(0,1) \leq PAR(t)$  then /* pitch adjustment for generation t */
08        $x'_i = x^{best}_k$ , where  $best$  is the index of best harmony in HM and
09          $k \sim U(1, N)$ 
10     end if
11   else /* random selection */
12      $x'_i = LB_i + r \times (UB_i - LB_i)$ 
13   end if
14 end for
15 Update HM /* replace the worst harmony */
16 until the NI is reached

```

Fig. 1. Pseudocode of the Global-Best Harmony Search algorithm (Adapted from [46])

adjustment step in HS so that the new harmony can mimic the best harmony in the harmony memory; this allows GBHS to work more efficiently in continuous, binary, and discrete problems than HS and IHS [46]. It was therefore selected as the basis for this proposal. A summary of the GBHS algorithm is shown in Fig. 1.

4 Algorithm Proposed for the Tagging Problem

4.1 Part-of-Speech Tagging (POST) Problem

A POST chooses a sequence of tags with the maximum probability for the set of words of a sentence [44]. To do this, Bayes interpretation considers all possible tag sequences [6], that is, it looks for a solution in T that maximizes Eq. 1 [13].

$$T^* = \arg \max \Pr(T|S) = \arg \max_{T \in \mathcal{T}} \left[\prod_{i=1}^n \Pr(w_i|t_i) \times \Pr(t_i|t_{i-1}) \right] \quad (1)$$

where T^* is the optimal solution with respect to all other possible solutions $\mathcal{T} = \{T^1, T^2, \dots, T^{N(n)}\}$, each $T^i = \{t^i_1, t^i_2, \dots, t^i_n\}$ represents a candidate tag, n is the number of words, in a sentence, $S = w_1, w_2, \dots, w_n$, (w_i indicates each word to be tagged), $\Pr(w_i|t_i)$ represents the probability of a word in a sentence given a tag and $\Pr(t_i|t_{i-1})$ represents the probability of a tag given the previous tag and the probability of the tag of the same word [13].

4.2 Part-of-Speech Tagging as an Optimization Problem

To represent the POST problem as an optimization problem it is necessary to keep in mind that the aim is to find the best tags for a sequence of words. The following considerations were taken into account:

- The possible tags of a word will be identified according to the tag set proposed by Petrov et al. [38]
- Next considerations are based on previous works [12, 13, 48]:
- Candidate solutions will be the most probable sequence of tags for the set of words to be tagged, which are obtained from the different tags that a word can have in view of its context. Each solution is represented as a vector of the size of the number of words in the sentence (each word is a position of the vector). In this work, an additional vector representing the cumulative probability of tagging each word, and its relation to its predecessor and successor was included. Finally, the solution includes the fitness function as the sum of the logarithms of these probabilities of each word, calculated according to Eq. 3. Figure 2 shows the representation of each solution or harmony.

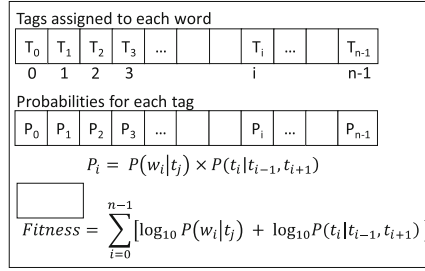


Fig. 2. Representation of the solution vector

- The context for selecting the most likely tag for a word will consider the window of a predecessor word and a successor word of the word to be tagged.
- Considering the two previous items, the fitness function includes the probability that each of the possible tags (t_j) of the word (w_i) to tag $P(w_i|t_j)$, which is independent of its context and the probability of the tags of the predecessor word and the successor word to that which it is desired to tag, given the tag of that word. The trigram will be evaluated as Eq. 2 as it is presented in [13].

$$Fitness = \prod_{i=1}^n P(w_i|t_j) \times P(t_i|t_{i-1}, t_{i+1}) \tag{2}$$

- To avoid the evaluation of the fitness function becoming zero after multiplying several probabilities close to zero, it was necessary to apply logarithms to each of the products of the objective function and make a sum of these, as shown in Eq. 3. In addition, whenever a trigram evaluated in the solution was not found in the training dataset, a default value of 0.000001 was assigned.

$$Fitness = \sum_{i=1}^n [\log_{10} P(w_i|t_j) + \log_{10} P(t_i|t_{i-1}, t_{i+1})] \tag{3}$$

4.3 Global-Best Harmony Search Tagger

Taking into account the fact that the GBHS algorithm performs better than the original version of HS, the possibility arose of using it to solve the tagging problem as an optimization problem, given the results presented in [13, 48] using HS as a base for the construction of a POST. The proposed GBHS Tagger algorithm involves the same parameters as its original version, such as HMCR, PARMIn, PARMMax, HMS and NI. It also involves two parameters such as: probability of optimization (ProbOpt), which controls the percentage of times that a local optimization process will be applied to the best harmony (solution) in harmony memory and the number of neighbors (MaxNeighbors) that will be evaluated in the local optimization process. Additionally, an Alpha parameter is included that controls whether the components of each harmony in the population are randomly generated from their possible labels or taken from the label with the greatest probability.

Figure 3 summarizes the structure of the GBHS Tagger, which involves local optimization to optimize the best harmony in the HM. The evaluation of the objective function involves the registration in a hash table of the solution vectors (harmonies) that have already been considered, like a tabu list and it's calculated with Eq. 3.

```

01 Define parameters such as HMS, NI, HCMR, PARMIn, ParMax, ProbOpt, Alpha,
    MaxNeighbors
02 Random initialization of HM or improved initialization using the Alpha parameter
03 for  $i = 1$  to NI do
04      $PAR \leftarrow PARMIn + (PARMax - PARMIn) \times (i/NI)$  ! definition of PAR *!
05     for  $j = 1$  to  $n$  do ! for each word in the sentence *!
06         if (Active[j] == true) then ! current word has more than one possible tag? *!
07             if ( $U(0,1) \leq HMCR$ ) then ! memory consideration *!
08                  $x'_j \leftarrow x_j^p$ , where  $p \sim U(1, \dots, HMS)$ 
09             if ( $U(0,1) \leq PAR$ ) then
10                  $x'_j \leftarrow x_k^{best}$ , where best is the index of best harmony in HM and
                     $k \sim U(1, n)$ 
11             end if
12         else ! random selection *!
13              $x'_j \leftarrow LB_j + r \times (UB_j - LB_j)$ 
14         end if
15     else
16          $x'_j \leftarrow \text{UniqueTagForWord}(j)$ 
17     end if
18 end for
19 while (visited ( $x'_j$ )) do ! the solution has already been visited *!
20     if (Active[j] == true) then mutate the new harmony ( $x'_j$ )
21         Randomly select a different one from the current one
22     end while
23 Evaluate the fitness of the new harmony ( $x'_j$ ) through the Eq.3
24 if (fitness of  $x'_j >$  fitness of worst harmony in HS) then
25      $HM[pos\_worst] \leftarrow x'_j$  ! replace the worst in harmony memory *!
26 end if
27 if ( $U(0,1) < ProbOpt$ ) then
28     Apply Local Optimization to the best harmony in HM
29 end if
30 end for
31 return best harmony in HM

```

Fig. 3. Pseudocode of proposed GBHS Tagger algorithm

In Fig. 4, the local optimizer used in line 28 of GBHS Tagger is detailed. The algorithm is an adapted version of Hill Climbing [43], which evaluates the maximum number of neighbors of the harmony. To define the neighbor, the word with the worst probability (P_i) of the harmony is selected and the tag assigned is changed to another possible value. If this new harmony is better than the original, the last one is replaced with the new one. The process is repeated but, to create the next neighbor, a different word must be selected, and so on. This constraint avoids over-exploitation of the same word in the harmony. Removing the word with the worst probability helps the optimizer find better solutions than using random selection (knowledge of the problem).

```

Hill Climbing (harmony current)
01  TabuListOfWords ← ∅
02  for i=1 to MaxNeighbors do
03      t ← index of the word with the lowest probability Pi that do not exist in the
04          TabuListOfWords and has more than one possible tag
05      if (t == ∅) then exit /* finalize the local optimization process */
06      TabuListOfWords.add(t) /* this word cannot be used again */
07      newHarmony ← Copy(current)
08      Change tag (randomly) of the word t in the newHarmony
09      if (Fitness(newHarmony) > Fitness(current)) then
10          current ← newHarmony
11      end if
12  end for

```

Fig. 4. Pseudocode of local Hill Climbing optimizer for GBHS Tagger algorithm

5 Experiments, Analyses, and Comparisons

5.1 Configuration

For the experiments, the Brown corpus [40] was used for English. This corpus is one of the most widely used for carrying out this type of experiments, consolidating 52,998 sentences, where each word was tagged with its corresponding universal tag [38]. The sentences of the corpus were divided into 5 folders, so that the tests could be performed using cross-validation. Table 1 shows the distribution of the sentences and words in each test and training data sets, for example, if the sentences of folder 1 are taken as test data, the training sentences are taken from folders 2 to 5 and so on for the other folders.

Table 1. Train and evaluation datasets used for the experiments.

Test data folder	Sentences in test data	Words in test data	Training data folders	Words in training data	Common words	Unknown words
1	10595	23105	2, 3, 4, 5	45113	18398	4707 (20.4%)
2	10600	22852	1, 3, 4, 5	45199	18231	4621 (20.2%)
3	10600	23130	1, 2, 4, 5	45009	18319	4811 (20.8%)
4	10600	22929	1, 2, 3, 5	45130	18239	4690 (20.5%)
5	10603	23111	1, 2, 3, 4	45025	18316	4795 (20.8%)

Next, the base probabilities for running the POST algorithms were calculated, i.e. the probability of each word and its possible tags in the different sets of information, as well as the probabilities of the trigrams.

The precision measure was then established that would be used for the performance evaluation of the algorithms with which the tests were performed (see Eq. 4) [13]:

$$\text{Precision} = \# \text{correctly tagged words} / \# \text{words} \quad (4)$$

Finally, two implementations of HSTAGger [13, 48] (which presents good results in contrast to recognized methods, Hidden Markov Models, Maximum Entropy, and a

version of Brill's model) and of a Random Tagger algorithm (generates new solutions randomly) were used for comparison with the proposed algorithm. The first implementation of HSTAGger corresponds to that proposed in [48] and the second one to that proposed in [13]. Comparison against GBHS Tagger was made without local optimization and with different probability optimization values (i.e. ProbOpt equal to 0.3, 0.5, and 0.7). Also, two versions of GBHS Tagger were built. The first version uses random initialization and the second one uses the Alpha parameter for the improved initialization of the harmony memory (GBHS Tagger 2).

Each algorithm was run 30 times over each sentence and the mean precision values and their standard deviation were calculated. All algorithms were run by performing a maximum of 110 evaluations of the objective function for each sentence.

The parameters used for HSTAGger were: HMS = 20, HMCR = 0.65 and PAR = 0.25. The parameters used for GBHS were: HMS = 10, HMCR = 0.95, PARMIn = 0.01, PARMMax = 0.99, ProbOpt = 0.0, 0.3, 0.5, 0.7 and MaxNeighbors = 5, Alfa = 0.5. A parameter tuning process must be performed since the values taken are those recommended in the paper of HSTAGger and those of GBHS Tagger recommended in GHS.

In addition, the NLTK software [49] and its data were used to execute the TnT tagger, using the Brown corpus as training and testing data.

5.2 Results

Table 2 shows the performance of the algorithms under the conditions described above. GBHS Tagger with local optimization (memetic version in both versions) presents better precision values than Random, TnT, HSTAGger, HSTAGger2 and GBHS Tagger without local optimization. This table also shows the performance of taggers in sentences containing unknown words observing that the proposed GBHS Tagger show better values of precision in both versions, but that version 2 obtained the best results.

Table 2. Results of running algorithms. Best results are shown in bold

Algorithms	Parameters	# of sentences	Precision (%)	Standard deviation	Precision (%) unknown words	Standard deviation
TnT	–	52998	83.5007	0.0659	80.9674	0.0225
Random	–	52998	82.4409	0.8313	79.9964	0.0074
HSTAGger	–	52998	91.5903	0.3232	88.8583	0.0028
HSTAGger2	–	52998	92.2751	0.0511	90.2307	0.0013
GBHS Tagger	0.0	52998	92.2474	0.3820	89.7843	0.0035
GBHS Tagger	0.3	52998	93.4417	0.2516	90.8924	0.0024
GBHS Tagger	0.5	52998	93.4444	0.2512	90.8959	0.0024
GBHS Tagger	0.7	52998	93.4414	0.2514	90.8923	0.0024
GBHS Tagger 2	0.0	52998	94.5615	0.0481	91.6568	0.0008
GBHS Tagger 2	0.3	52998	95.1959	0.0314	92.8542	0.0005
GBHS Tagger 2	0.5	52998	95.1959	0.0314	92.8542	0.0005
GBHS Tagger 2	0.7	52998	95.1959	0.0314	92.8542	0.0005

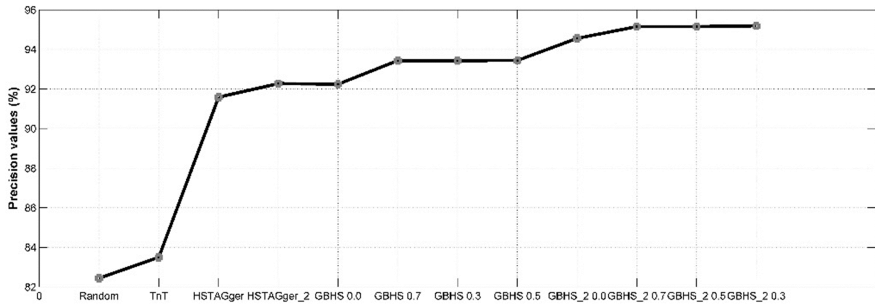


Fig. 5. Comparison of GBHS Tagger algorithms

In Fig. 5, the performance of the precision values for each one of the evaluated algorithms can be appreciated. A significant difference between the precision values of the algorithms can be seen, confirming that the GBHS Tagger2 algorithm performs better.

Table 3. Precision obtained by folder. Best results are shown in bold

Algorithm	Folders					Total	Stdev	Avg. time (seconds)
	1	2	3	4	5			
GBHS Tagger 2 with 0.3	95.2239	95.1787	95.2025	95.2299	95.1444	95.1959	0.0314	25.2849
GBHS Tagger 2 with 0.5	95.2239	95.1787	95.2025	95.2299	95.1444	95.1959	0.0314	24.80547
GBHS Tagger 2 with 0.7	95.2239	95.1787	95.2025	95.2299	95.1444	95.1959	0.0314	24.800,8
GBHS Tagger 2 with 0.0	94.6291	94.4980	94.5209	94.5980	94.5615	94.5615	0.0481	23.7773
GBHS Tagger with 0.5	93.6805	93.0410	93.2689	93.5547	93.6768	93.4444	0.2512	12.0091
GBHS Tagger with 0.3	93.6769	93.0373	93.2665	93.5519	93.6758	93.4417	0.2516	11.4179
GBHS Tagger with 0.7	93.6769	93.0373	93.2665	93.5519	93.6747	93.4414	0.2514	10.6113
GBHS Tagger with 0.0	92.6221	91.6444	91.9596	92.4364	92.5747	92.2474	0.3820	10.9308
HSTAGger 2	92.2904	92.3189	92.2742	92.3140	92.1782	92.2751	0.0511	22.3344
HSTAGger	91.9486	91.0889	91.3380	91.8070	91.7693	91.5903	0.3232	12.0829
TnT	83.5415	91.9662	78.8870	83.3974	83.5573	83.5007	0.0659	13,7691
Random Tagger	83.3960	81.2045	81.7253	82.8884	82.9901	82.4409	0.8313	11,7691

Table 3 shows the precision values for each algorithm and for each folder. For all test sentences the GBHS Tagger algorithms reports better precision values than the Random and HSTAGger algorithms, obtaining their best value when the probability of local optimization is 0.5 for GBHS Tagger and 0.2 for GBHS Tagger 2. It can also be seen from this table that the precision values are similar in each algorithm regardless of the folder that has been used. This can be seen with the last row that shows the standard deviation of each algorithm and its precision data by folder. The Random algorithm is the one with the highest deviation and GBHS Tagger 2 algorithms with optimization and improved initialization those that show the least deviation or most consistent precision results.

The Friedman test was applied to these data to establish the differences between the algorithms, obtaining the scores shown in Table 4. In this table, we can see that the best tagging algorithm is GBHS Tagger 2 with a local optimization (of 0.3, 0.5 and

0.7). This test has 11 degrees of freedom (53.215385) and a p-value of $1.641E-7$. The results of GBHS Tagger 2 with a probability of optimization from 0.3 onwards are better than those without probability for the same algorithm and better than the GBHS Tagger algorithm (version 1) for all values of optimization. HSTAGger2 is better than HSTagger and TnT and finally, TnT is better than Random Tagger. These results also make it possible to appreciate that the use of improved initialization of the harmony memory (GBHS Tagger 2 algorithm) makes local optimization impact the performance of the algorithm only until a value of 0.3, thereafter, the algorithm does not show performance improvements. The performance of TnT is affected by the manual tokenization process of the Brown corpus whereas the meta-heuristic proposals do not. For example, in the sentence "... the recent atlanta's investigation...", TnT label the token "atlanta's" wrongly, but if this token is divided in two: "atlanta" and "s", TnT label each token correctly. The same situation affects the performance of other tagger in NLTK such as HMM and Perceptron.

Table 4. Friedman ranking

#	Algorithm	Ranking	#	Algorithm	Ranking
1	GBHS Tagger 2 with 0.3	2	7	GBHS Tagger with 0.7	6.6
2	GBHS Tagger 2 with 0.5	2	8	GBHS Tagger with 0.0	8.6
3	GBHS Tagger 2 with 0.7	2	9	HSTAGger 2	8.6
4	GBHS Tagger 2 with 0.0	4	10	HSTAGger	10.2
5	GBHS Tagger with 0.5	5	11	TnT	10.8
6	GBHS Tagger with 0.3	6.4	12	Random Tagger	11.8

6 Conclusions and Future Work

GBHS Tagger algorithm (with or without improvement in the initialization of the harmony memory) is a proposal for a POST from the perspective of an optimization problem that finds the best combination of tags for a set of words in a sentence, obtaining outstanding results in its performance compared to the other algorithms evaluated. GBHS Tagger includes an additional parameter that controls the probability of local optimization, achieving the best results for the first version when the probability is 0.5 (with no improvement in the initialization of the harmony memory) and for GBHS Tagger 2 when the probability is of 0.3 onwards (using improved initialization of HM). The experiments used the sentences of the Brown corpus for English and were divided into 5 folders to evaluate the algorithms using cross validation.

As future work, the research group hopes first to make improvements in the objective function and the way in which the local optimization is done, given that including knowledge in that optimizer will potentiate the performance of the algorithm; secondly, to develop a complete tuning process of the parameters for the proposed algorithm; thirdly, to propose and evaluate other metaheuristic algorithms for POST problem such as Differential Evolution hybridized with k-means [50] and PSO [51]; and finally, to use the GBHS Tagger algorithm over other traditional and non-traditional languages, as well as other English corpora.

Acknowledgements. Sierra, Cobos and Corrales are grateful to University of Cauca and its research groups GTI and GIT of the Computer Science and Telematics departments. We are especially grateful to Colin McLachlan for suggestions relating to the English text.

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: *Modern Information Retrieval*. Pearson - Addison Wesley, New York (1999)
2. Sammut, C., Webb, G.I. (eds.): *Encyclopedia of Machine Learning (Part of Speech Tagging)*. Springer, New York (2010)
3. Paul, A., Purkayastha, B.S., Sarkar, S.I.: Hidden Markov model based part of speech tagging for Nepali language. In: 2015 International Symposium on Advanced Computing and Communication (ISACC), Silchar, pp. 149–156 (2015)
4. Makazhanov, A., Yessenbayev, Z., Sabyrgaliyev, I., Sharafudinov, A.: On certain aspects of Kazakh part-of-speech. In: IEEE 8th International Conference on Application of Information and Communication Technologies (AICT), Astana, pp. 1–4 (2014)
5. Ratnaparkhi, A.: A maximum entropy model for part-of-speech tagging. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 113–142 (1996)
6. Ariaratnam, I., Weerasinghe, A.R., Liyanage, C.: A shallow parser for Tamil. In: 2014 International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, pp. 197–203 (2014)
7. Brants, T.: TnT - a statistical part-of-speech tagger. In: *Proceedings of the Sixth Conference on Applied Natural Language Processing, ANLC 2000*, Stroudsburg, PA, USA, pp. 224–231 (2000)
8. Brill, E.: A simple rule-based part of speech tagger. In: *Proceedings of the Third Conference on Applied Natural Language Processing, ANLC 1992*, Stroudsburg, PA, USA, pp. 152–155 (1992)
9. Brill, E.: Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. *Comput. Linguist.* **21**(4), 543–565 (1995)
10. Schmid, H.: Part-of-speech tagging with neural networks. In: *Proceedings of the 15th Conference on Computational Linguistics*, Stroudsburg, PA, USA, pp. 172–176 (1994)
11. Nakamura, M., Shikano, K.: A study of English word category prediction based on neural networks, acoustics, speech, and signal processing. In: *International Conference on Acoustics, Speech, and Signal Processing, IEEE, Glasgow*, pp. 731–734 (1989)
12. Forsati, R., Shamsfard, M., Mojtahedpour, P.: An efficient meta heuristic algorithm for POS-tagging. In: 2010 Fifth International Multi-Conference on Computing in the Global Information Technology (ICCGI), Valencia (2010)
13. Forsati, R., Shamsfard, M.: Novel harmony search-based algorithms for part-of-speech tagging. *Knowl. Inf. Syst.* **42**(3), 709–736 (2015)
14. Silva, A.P., Silva, A., Rodríguez, I.: An approach to the POS tagging problem using genetic algorithms. In: Madani, K., Correia, A., Rosa, A., Filipe, J. (eds.) *Computational Intelligence. Studies in Computational Intelligence*, vol. 577, pp. 3–17. Springer, Cham (2012). https://doi.org/10.1007/978-3-319-11271-8_1
15. Jianchao, T.: An English part of speech tagging method based on maximum entropy. In: 2015 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Halong Bay, Vietnam, pp. 76–80 (2015)
16. Ranjan Das, B., Sahoo, S., Sekhar Panda, C., Patnaik, S.: Part of Speech tagging in Odia using support vector machine. In: *Procedia Computer Science, International Conference on Intelligent Computing, Communication Converge, ICCO-2015*, vol. 48, pp. 507–512 (2015)

17. Ekbal, A., Bandyopadhyay, S.: Part of speech tagging in Bengali using support vector machine. In: International Conference on Information Technology, ICIT 2008, pp. 10–111 (2008)
18. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th International Conference on Machine Learning 2001, pp. 282–289 (2001)
19. Araujo, L.: How evolutionary algorithms are applied to statistical natural language processing. *Artif. Intell. Rev.* **28**(4), 275–303 (2007)
20. AlSuhaibani, R.S., Newman, C.D., Collard, M.L., Maletic, J.I.: Heuristic-based part-of-speech tagging of source code identifiers and comments. In: 2015 IEEE 5th Workshop on Mining Unstructured Data (MUD), Bremen, pp. 1–5 (2015)
21. Aziz, T.A., Sunitha, C.: A hybrid parts of speech tagger for Malayalam. In: 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, pp. 1502–1507 (2015)
22. Mall, S., Jaiswal, U.C.: Innovative algorithms for parts of speech tagging in Hindi-English machine. In: 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), Noida, pp. 709–714 (2015)
23. Tian, Y., Lo, D.: A comparative study on the effectiveness of part-of-speech tagging techniques on bug reports. In: 2015 IEEE 22nd International Conference on Software Analysis, Evolution and Reengineering (SANER), Montreal, QB, pp. 570–574 (2015)
24. Carneiro, H.C., França, F.M., Lima, P.M.: Multilingual part-of-speech tagging with weightless neural networks. *Neural Netw.* **66**, 11–21 (2015)
25. Carneiro, H.C., França, F.M., Lima, P.M.: WANN-tagger - a weightless artificial neural network tagger for the Portuguese language. In: Proceedings of the International Conference on Fuzzy Computation and International Conference on Neural Computation, ICFC-ICNC 2010, Valencia, pp. 330–335 (2010)
26. Poel, M., Boschman, E., Akker, R.A.: Neural network based Dutch part of speech tagger. In: Proceedings of the Twentieth Belgian-Dutch Artificial Intelligence Conference, BNAIC 2008, The Netherlands, pp. 217–224 (2008)
27. Zennaki, O., Semmar, N., Besacier, L.: Unsupervised and lightly supervised part-of-speech tagging using recurrent neural networks. In: 29th Pacific Asian Conference on Language, Information and Computation, Shanghai, China, pp. 133–142 (2015)
28. Duong, L., Cohn, T., Verspoor, K., Bird, S., Cook, P.: What a can we get from 1000 tokens? A case study of multilingual POS tagging for resource-poor languages. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, pp. 886–897 (2014)
29. Forsati, R., Shamsfard, M.: Cooperation of evolutionary and statistical statistical POS-tagging. In: 2012 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP), Shiraz, Fars, pp. 446–451 (2012)
30. Silva, A.P., Silva, A., Rodríguez, I.: Part-of-speech tagging using evolutionary computation. In: Terrazas, G., Otero, F., Masegosa, A. (eds.) *Nature Inspired Cooperative Strategies for Optimization (NICSO 2013)*. Studies in Computational Intelligence, vol. 512, pp. 167–178. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-01692-4_13
31. Silva, A.P., Silva, A., Rodríguez, I.: Tagging with disambiguation rules a new evolutionary approach to the part-of-speech tagging problem. In: Proceedings of the 4th International Joint Conference on Computational Intelligence, ECTA-2012, pp. 5–14 (2012)
32. Silva, A.P., Silva, A., Rodríguez, I.: A new approach to the POS tagging problem using evolutionary. In: Proceedings of Recent Advances in Natural Language Processing, Hissar, Bulgaria, pp. 619–625 (2013)

33. Silva, A.P., Silva, A., Rodríguez, I.: PSO-tagger: a new biologically inspired approach to the part-of-speech tagging problem. In: Tomassini, M., Antonioni, A., Daolio, F., Buesser, P. (eds.) ICANNGA 2013. LNCS, vol. 7824, pp. 90–99. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37213-1_10
34. Bachir Menai, M.E.: Word sense disambiguation using evolutionary algorithms – application to Arabic language. *Comput. Hum. Behav.* **41**, 92–103 (2014)
35. Ekbal, A., Saha, S.: Simulated annealing based classifier ensemble techniques: application to part of speech tagging. *Inf. Fusion* **14**(3), 288–300 (2013)
36. Ekbal, A., Saha, S.: A multiobjective simulated annealing approach for classifier ensemble: named entity recognition in Indian languages as case studies. *Expert Syst. Appl.* **38**, 14760–14772 (2011)
37. Dinakaramani, A., Rashel, F., Luthfi, A., Manurung, R.: Designing an Indonesian part of speech tagset and manually tagged Indonesian corpus. In: 2014 International Conference on Asian Language Processing (IALP), Kuching (2014)
38. Petrov, S., Das, D., McDonald, R.: A universal part-of-speech tagset. In: Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012, Istanbul (2012)
39. Rabbi, I., Khan, M.A., Ali, R.: Developing a tagset for Pashto part of speech tagging. In: Second International Conference on Electrical Engineering, Lahore (Pakistan) (2008)
40. Francis, W.N., Kucera, H.: Brown Corpus (1979). <http://clu.uni.no/icame/manuals/BROWN/INDEX.HTM#bc8>. Accessed 21 Nov 2016
41. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: the penn treebank. *J. Comput. Linguist. Spec. Issue Using Large Corpora, II* **19**(2), 313–330 (1993)
42. Koehn, P.: Europarl: a parallel corpus for statistical machine translation. In: Proceedings of the Tenth Machine Translation Summit (MT Summit XX), Phuket, Thailand (2005)
43. Brownlee, J.: Clever algorithms nature-inspired programming recipes. Melbourne, lulu.com (2011)
44. Yang, X.-S.: Harmony search as a metaheuristic algorithm. In: Geem, Z.W. (ed.) Music-Inspired Harmony Search Algorithm. Studies in Computational Intelligence, vol. 191, pp. 1–14. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00185-7_1
45. Lee, K.S., Geem, Z.W.: A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Comput. Methods Appl. Mech. Eng.* **194**, 3902–3933 (2005)
46. Mahdavi, M., Fesanghary, M., Damangir, E.: An improved harmony search algorithm for solving optimization problems. *Appl. Math. Comput.* **188**(2), 1567–1579 (2007)
47. Omrán, M.G., Mahdavi, M.: Global-best harmony search. *Appl. Math. Comput.* **198**, 643–656 (2008)
48. Forsati, R., Shamsfard, M.: Hybrid PoS-tagging: a cooperation of evolutionary and statistical approaches. *Appl. Math. Model.* **38**(13), 3193–3211 (2014)
49. NLTK Project: Natural Language Toolkit (2017). <http://www.nltk.org/>. Accessed 15 June 2017
50. Sierra, L.-M., Cobos, L., Corrales, J.-C.: Continuous optimization based on a hybridization of differential evolution with k-means. In: Bazzan, A.L.C., Pichara, K. (eds.) IBERAMIA 2014. LNCS, vol. 8864, pp. 381–392. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12027-0_31
51. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micromachine and Human Science (1995)

A Study on Crossmodal Correspondence in Sensory Pathways Through Forced Choice Task and Frequency Based Correlation in Sound-Symbolism

Keerthi S Chandran^{1(✉)}, Swati Banerjee², and Kuntal Ghosh^{1,2}

¹ Center for Soft Computing Research, Indian Statistical Institute,
203 B T Road, Kolkata 700108, India

keerthischandran@gmail.com, kuntal@isical.ac.in

² Machine Intelligence Unit, Indian Statistical Institute, 203 B T Road, Kolkata 700108, India
swatibanerjee@ieee.org

Abstract. The crossmodal correspondence in sensory pathways of human can get revealed by subjecting them to forced choice task as in sound-symbolism. Sound-symbolism is a term used for a hypothetical systematic relationship between word and meaning. A well known case in sound symbolism is the Kiki-Bouba phenomenon in which a subject labels a jagged figure as Kiki and rounded figure as Bouba when presented with both figures and words and asked to label the figure with the words. In the current experiment the words for cotton and sword were chosen from foreign languages and the subjects were asked to label the figure with that pair. Majority of subjects labeled the pointed figure with the word for sword for most of the languages. The word for sword had higher frequency components in most languages. The subjects may be associating words with higher frequency components to the jagged figure which implies possible crossmodal correspondence between visual and auditory pathways as was also indicated by neuropsychologists for natural language understanding.

Keywords: Sound-symbolism · Phonology · Cognitive linguistics · Semantics · Multisensory perception · Natural language understanding

1 Introduction

Sound-symbolism is a term used for a hypothetical systematic relation for word and meaning. The concept of sound-symbolism goes as far as Plato who records the conversation between two philosophers, Socrates and Hermogenes. Hermogenes argued that all names are conventions and habits of users while Socrates argued that names by nature have a truth in them [1, 2]. Mainstream linguists today hold the view of Ferdinand de Saussure that the connection between words and meaning in languages are arbitrary. However there is also enough interest and research in the field of sound-symbolism. Saussure's contemporary Otto Jespersen surveyed the word for little, "child or young animals", "small things", and diminutive suffixes and found that that vowel 'i' is used cross linguistically to denote the things that are "small slight and insignificant" [3].

Sapir performed an experiment following this in which 74.6%–96.4% of participants labeled an object of bigger size *mal* and smaller size *mil* when they were asked to label the two using the same words. Wolfgang Köhler performed a similar experiment in which majority of Spanish speaking subjects labeled a rounded/curvy shape *maluma* and an angular shape *takate*. This phenomenon received considerable attention when Ramachandran and Hubbard reported that majority of subjects label a rounded shape as *Bouba* and jagged shape as *Kiki* when they were asked to do so [4].

The analysis of these relations will be useful in several areas of natural language processing [5]. Human subjects are known to perform better than chance in picking up the correct adjectives like *big/small*, *loud/quiet* in unknown foreign languages with those having synesthesia performing better than the others. Synesthesia is a condition in which one sensory input can evoke other sensations [6]. A statistically significant portion of non-Himbusan language speakers were able to perform better than chance when asked to identify between bird names and fish names in Himbusian language [7]. Modeling and understanding how the brain accomplishes such tasks can help in natural language processing.

In the present study, words for naturally angular objects like “sword” and blunt objects like “cotton” from languages of various origins are studied to learn about the effect of sound-symbolism based forced choice tasks and their relationship in the cross-modal correspondence in sensory pathway. The theory section gives an overview of the phenomenon followed by the material and method section where the data preparation and analysis is described. Here, the subjects were shown the figures used by Ramachandran and Hubbard [8] and were asked to label them using the word pairs given to them. For large majority of the cases the jagged object was labeled to sword and the rounded one to that of cotton. An attempt has been made to deduce ways by which a machine could guess between the words for sword and cotton from the analysis of their frequency spectrum. It was found that the word for sword had higher frequency components than that of cotton in most of the languages.

2 Theory

The exact nature of sound-symbolism is unknown where Ramchandran and Hubbard have suggested that the phenomenon involves synesthesia like cross connections in the brain [8]. The sharp sounds in *kiki* could be mimicking the sharp inflection of tongue in the palate and the *boobaa* sound could be mimicking the curves via the smooth rounding and curving of the lips [9]. Others like Ohala have suggested that that acoustic features of sound themselves could be the driving force [10].

It is known that different regions of basilar of cochlea vibrate in resonance to different frequencies [11]. The higher frequency components of sound wave could lead to transduction of neural impulses encoding higher frequencies. In case the frequency components of the words do play a part in the subjects’ decision in the forced choice task, we can analyze them using fast fourier transform (FFT).

The words that bear some form of symbolism with their meaning can be more easily learned and remembered and can thus become more common in vocabulary. The crossmodal connections could also have initiated the early evolution of language [9].

3 Materials and Methods

3.1 Participants

A total of fourteen subjects (6 males, 8 females, age 20–30) volunteered for the experiment. The subjects were native Bengali or Hindi speakers and did not have any prior knowledge of the languages used.

3.2 Stimuli

The words for sword and cotton in twenty four languages were downloaded from <https://www.collinsdictionary.com/>. In addition the Maori words for cotton ‘miro’ and sword ‘hoari’ in Maori dictionary were obtained from Maori dictionary at <http://maoridictionary.co.nz/>. Maori was selected because in this language the word ‘sword’ was included in Maori dictionary much later for historical reasons, which has been discussed later in Sect. 4. The words for sword downloaded had a mean duration of 0.75 s and the words for cotton had a mean duration of 0.763 s. European and Brazilian versions of Spanish and Portuguese were treated as different languages. Though they all had similar words they were pronounced by different speakers in slightly different manner. The subjects were asked to assign the words as names for two figures (Fig. 1).

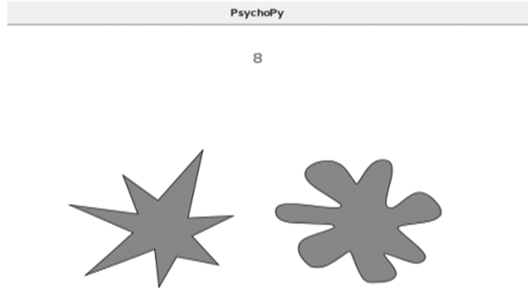


Fig. 1. The visual stimulus presented to subjects on computer screen.

3.3 Procedure

PsychoPy (version 1.84.2) is an open source psychophysics toolkit for python was used for displaying the stimulus and experiment design. The words were played twice to the subject via a computer speaker when they were looking into the above stimulus on the computer screen as shown in Fig. 1. A number was displayed for each language and the subjects were instructed to write down the words that they felt to correspond to each

shape along with the number on paper. The subjects could play the sounds again by pressing ‘r’ button on the keyboard. They could go to next set of sounds by pressing ‘n’. The subjects were not informed of the origin of the words.

3.4 Analysis of Words

The sound waves were subjected to a Fourier transformation in the amplitude space. The frequencies of different words thus transformed had different sets of frequencies. To make the frequencies used in FFT transformation of all words used the same; the corresponding amplitudes were interpolated to a frequency range of 0 to 20000 Hz in steps of one hertz. The maximum frequency was capped at 20000 Hz as the human audible frequency is between 20 Hz to 20000 Hz. The interpolation was done with `numpy.interp` function in the python library. A parameter high frequency fraction for a word was calculated for a particular frequency by summing up the amplitudes for frequencies above that particular frequency f and then dividing them by the total sum of amplitudes of all frequencies for that word. Then the ratio of high frequency fraction of sword to cotton in a language was calculated.

The high frequency fraction for a word at f hertz has been calculated as follows. The amplitude at a particular frequency obtained by FFT has been denoted as Amplitude (frequency).

$$High\ Frequency\ Fraction\ (f) = \frac{\sum_f^{20000} Amplitude\ (frequency)}{\sum_0^{20000} Amplitude\ (frequency)}$$

The high frequency fraction ratio (HFFR) for a language at a particular frequency has been calculated as:

$$High\ frequency\ fraction\ ratio\ (f) = \frac{frequency\ fraction\ of\ word\ for\ sword\ at\ f}{high\ frequency\ fraction\ for\ word\ for\ cotton\ at\ f}$$

4 Results and Discussion

The median number of languages in which the words for sword were matched to the jagged figure by subjects was 17.5 (Fig. 2). It can be seen by visual inspection that the word for sword has more elements in frequencies above about 2000 Hz in most of the words (Fig. 3). One of the exceptions was the Maori language (Fig. 4). Only two of the seven people who matched the word for sword to cotton in 18 or more languages, labeled the Maori word for sword to the jagged figure.

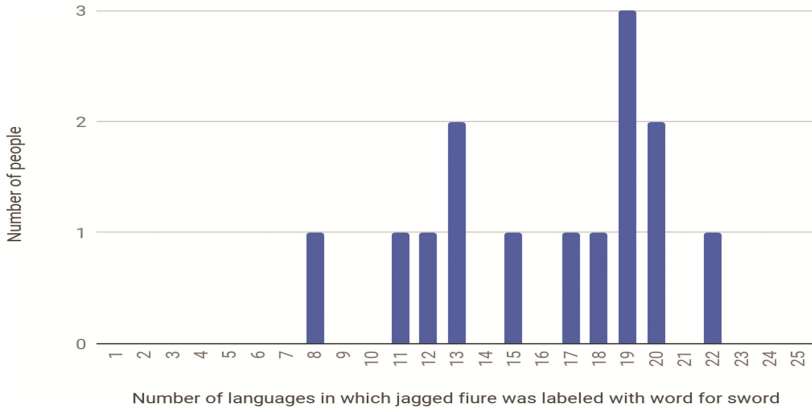


Fig. 2. The number of languages in which a particular person mapped the word for sword to round

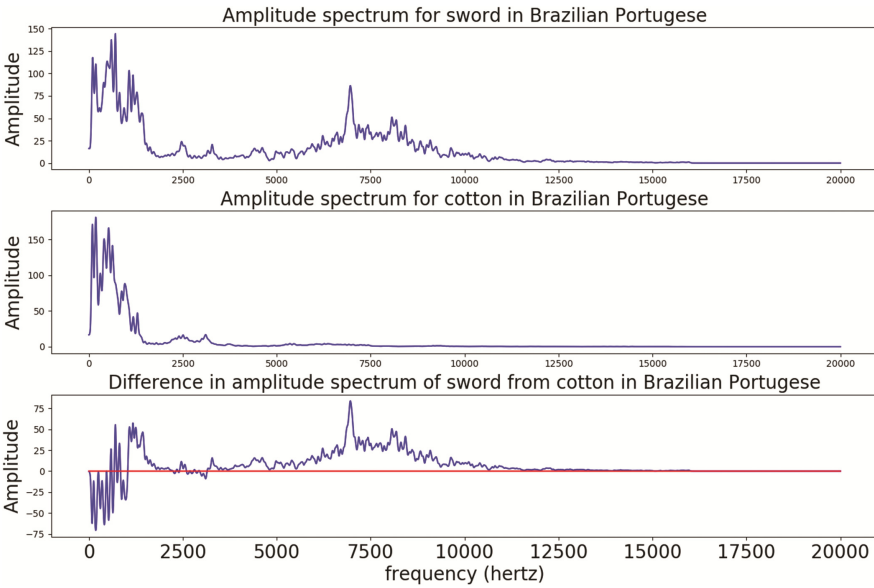


Fig. 3. The amplitude spectrum for the words denoting sword and cotton in Brazilian Portuguese. The difference of their amplitude spectrum has also been plotted. Curve has been smoothed with a Gaussian filter of sigma 20.

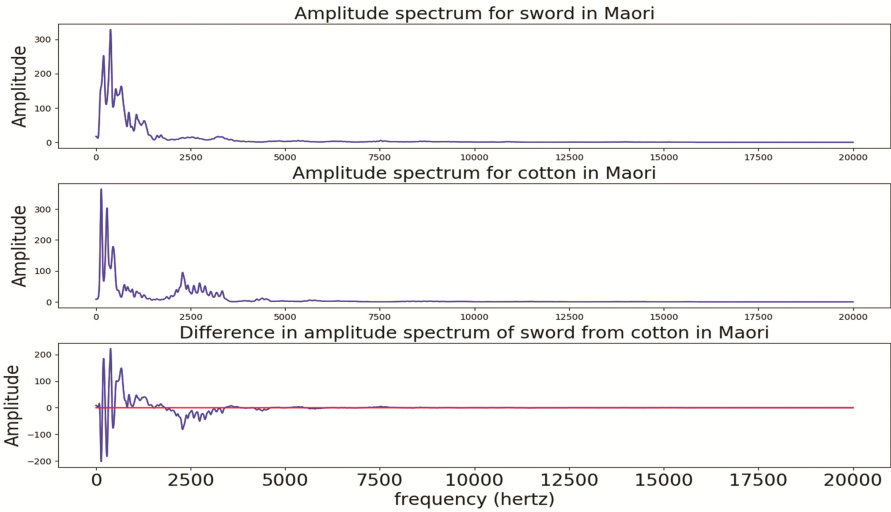


Fig. 4. The amplitude spectrum for the words denoting sword and cotton in Maori language. The difference of their amplitude spectrum has also been plotted. Curve has been smoothed with a Gaussian filter with sigma 20

It was found that the high frequency component of the word for sword was higher than cotton for most languages in most frequency ranges. Twenty three of twenty five languages had a high frequency fraction greater than one in the range of 260–379 Hz and 950 Hz to 2589 Hz (Fig. 5). The sum of all the amplitudes for frequencies from 0 to 20000 Hz was greater for the word for sword than cotton in nineteen languages. Only two languages had a high frequency fraction lesser than one in the 950 Hz to 2589 Hz range were Korean and Maori (Fig. 6). The languages in which the seven subjects who got maximum correct labeled the jagged figure with word for sword had a high frequency

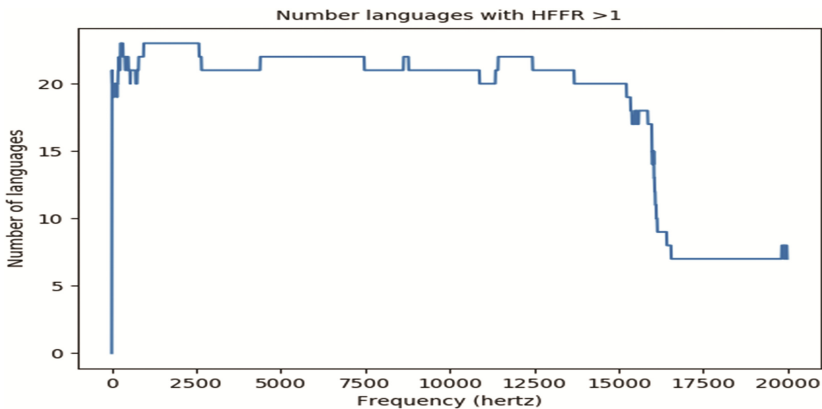


Fig. 5. The number of languages for which the ratio of high frequency ratio was more than one at various frequencies.

ratio above 1 near 950 Hz to 2589 Hz range, four of which have been plotted (Fig. 7). Of the 7 subjects who matched word of sword to the jagged figure in eighteen or more languages, five people had marked the Maori word for cotton to the jagged figure.

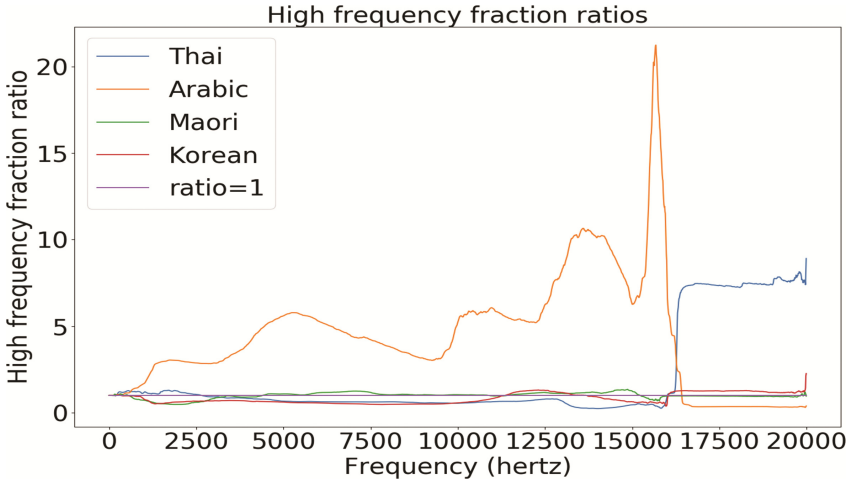


Fig. 6. The high frequency fraction ratio of four languages in which seven subjects above median did not match the word for sword with jagged figure. Number of matchings was 3 for Arabic, 2 for Maori, 4 each for Korean and Thai.

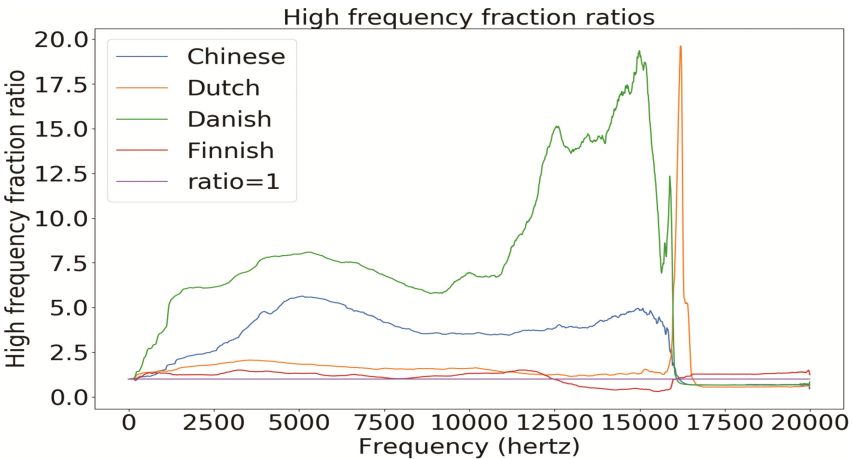


Fig. 7. The high frequency fraction ratio of four languages in which all subjects matched 18 or more words for sword with jagged figure (4 languages are shown for better visibility).

The Maoris were Polynesian islanders who settled in New Zealand without metallurgy. The Maori weapons did not include the sword and they quickly adopted muskets after the start of trade relations with Europeans [12, 13]. This could have prevented the

evolution of a word with higher frequency components for sword for Maori unlike other languages.

Majority of languages were found to have a higher frequency ratio for sword to cotton greater than one and most people matched the word for sword with the jagged figure. Spearman's rank correlation was calculated for the number of people who matched the word for sword to the jagged figure in various languages to ratio of high frequency fraction at various frequencies (Fig. 8). As, the study population is small in size, subjects scoring a value above the median were considered for analysis. A maximum correlation coefficient of 0.46 was obtained at around 2500 Hz.

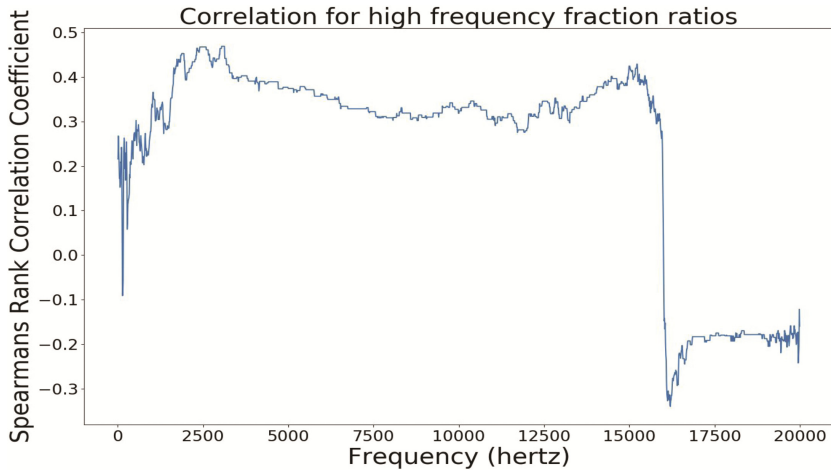


Fig. 8. The Spearman's rank correlation coefficient of the ratio of high frequency component fraction of sword and cotton in various languages to the number of matchings of sword to jagged figure in those languages obtained by people who did above median matchings at various frequencies

5 Conclusions

The angular shape of the sword and non angularity of cotton could have facilitated the evolution of the word for sword and cotton in different languages. The subjects have a tendency to match the word for sword thus evolved to a jagged shape than a rounded one when presented with a forced choice task. The words for sword generally have a higher frequency fraction than those of cotton in same language. The acoustic features of the word including their frequency components would have a role to play in choice of subjects' pairing of the words with the shapes in forced choice tasks. The subjects associate higher frequency components in the word with the jagged figure implying possible cross-modal correspondence between visual and auditory pathways in the brain as was also indicated by neuropsychologists [9]. Further investigations with a larger number of subjects can be made by digitally manipulating components and segments in the sound files and then performing forced choice tasks.

References

1. Ohala, J.H.: Sound Symbolism. <http://www.linguistics.berkeley.edu/~ohala/papers/SEOUL4-symbolism.pdf>
2. Hinton, L., Nichols, J., Ohala, J.J.: Introduction: Sound-Symbolic Processes, Sound Symbolism. Cambridge University Press, Cambridge (2006)
3. Jespersen, O.: Selected Writing of Otto Jespersen. Taylor and Francis, Hoboken (2010)
4. Spence, C.: Crossmodal correspondences: a tutorial review. *Atten. Percept. Psychophys.* **73**(4), 971–995 (2011). <https://doi.org/10.3758/s13414-010-0073-7>
5. Igarashi, T., Sasan, R., Takamura, H., Okumura, M.: 自然言語処理, vol. 20, no. 2, pp. 183–200 (2013). <http://doi.org/10.5715/jnlp.20.183>
6. Bankieri, K., Simner, J.: What is the link between synaesthesia and sound symbolism? *Cognition* **136**, 186–195 (2015). <https://doi.org/10.1016/j.cognition.2014.11.013>
7. Berlin, B.: Evidence for pervasive synesthetic sound symbolism in ethnozoological nomenclature. In: *Sound Symbolism*, pp. 76–93. <https://doi.org/10.1017/cbo9780511751806.006>
8. Ramachandran, V.S., Hubbard, E.M.: Synaesthesia—a window into perception, thought and language. *J. Conscious. Stud.* **8**(12), 3–34 (2001)
9. Ramachandran, V.S.: *The Tell-Tale Brain: A Neuroscientist's Quest for What Makes us Human*. W.W. Norton, New York (2012)
10. Ohala, J.J.: Speech perception is hearing sounds, not tongues. *J. Acoust. Soc. Am.* **95**(5), 2849 (1994). <https://doi.org/10.1121/1.409549>
11. Rebillard, G., Pujol, R.: <http://www.cochlea.eu/en/cochlea/function>
12. Schwimmer, E.G.: Warfare of Maori. *Te Ao Hou (The New World)* **36**, 51–54 (1961)
13. Diamond, J.M.: *Guns, Germs and Steel: A Short History of Everybody for the Last 13,000 Years*. Vintage, London (2005)

Point Process Modeling of Spectral Peaks for Low Resource Robust Speech Recognition

Anupam Mandal¹(✉), K. R. Prasanna Kumar¹, and Pabitra Mitra²

¹ Center for AI and Robotics, Bangalore, India
{amandal, prasanna}@cair.drdo.in

² Department of CSE, IIT Kharagpur, Kharagpur, India
pabitra@cse.iitkgp.ernet.in

Abstract. The paper proposes an approach for noise robust speech recognition in low resource setting. The approach involves formulation of whole word point process model based on word specific spectral peak event in selected groups of mel banks. The performance of the proposed approach is demonstrated on an isolated word recognizer on noisy speech samples (additive white Gaussian noise) at different *SNR* levels ranging from 0 dB to clean speech. The training is carried out with examples varying from 5 to 80. Performance comparison with HMM based system trained with mel-frequency cepstral coefficients (MFCC) features show an improvement of 8–17% (absolute) depending on *SNR* level when the number of training examples are less than 10. Since the approach relies only on positions and magnitudes of spectral peaks derived from spoken word examples without any language specific resources, the same can potentially be applied for any language. It is also shown that our approach recognizes those words better that are poorly recognized by HMMs across all *SNR* levels.

Keywords: Speech recognition · Low resource · Spectral peaks
Point process modeling

1 Introduction

The state-of-the art techniques of speech recognition employing Hidden Markov Models [8] and Deep Neural Networks [1] are often of limited use in low resource setting due to their requirements of statistically significant amounts of training resources. These techniques are able to learn the various model parameters faithfully when presented with adequate training examples thereby demonstrating high recognition accuracies under matched conditions. But in real life scenarios it may not always be possible to get large number of training examples either because of limited training resources as in the case of low resource languages [3] or due to operational constraints such as availability of a few training templates in a query-by-example framework. The difficulty gets compounded with introduction of noise in speech in addition to limited number of available examples

that results in steep decrease in recognition accuracies. This motivates exploring alternate representation that will be robust to additive noise. It is seen that spectral peaks obtained framewise from mel-scale weighted magnitude spectrogram of a speech utterance remain preserved in presence of additive stationary (Gaussian) noise. This is unlike mel-frequency cepstral coefficients (MFCC) that are known to get affected significantly. The spectral peaks exhibit sparse temporal patterns that can be considered as a collection of independent point patterns. In view of the sparseness of representation, point process modeling (PPM) of such pattern as an inhomogenous Poisson process is proposed. This is because PPM is known to exhibit a natural resilience to non-stationary noise [5,6] apart from being suitable for temporal modeling of sparse events [2]. We build a isolated word recognizer, combining our spectral peak representation with point process modeling. We evaluate its performance on stationary noisy (additive Gaussian noise) speech data derived from TI46 [11] database at various noise levels and limited number of training examples. However, there are significant differences of our approach from [2,6,7,9], primarily in sparse event representation and formulation of PPMs. While all of the above use acoustic event detectors (variants of phone recognizers), our approach uses spectral peak detectors for sparse event representation. The acoustic event detector uses a phone recognizer trained on clean speech that is sensitive to the presence and distribution of noise and requires adequate training resources. Our proposed approach is different in that it does not depend on phone recognizer and exhibits robust performance on noisy speech with limited number of examples. It makes use of both peak position and amplitude while the above approaches use only the positions of acoustic event position and not their strength. The approach is also language independent as the spectral peaks are derived purely from the signal and does not make an assumption of an underlying language. Our approach is also different from [4] that considers peaks from all filter banks for PPM formulation while our approach focuses on word specific groups of mel banks. We demonstrate this using an isolated word recognizer trained on TI46 database with various number of examples in presence of various noise levels. This establishes the better suitability of our proposed approach compared to HMMs for robust speech recognition in a low resource setting under low *SNR* conditions.

The rest of the paper is organized as follows. We begin with a description of the baseline HMM recognizer used in this study in Sect. 2. The point process representation of spectral peaks is discussed in Sect. 3, followed by that of bank-wise segmentation of the spectral peaks in Sect. 4. The PPM based isolate word recognizer is presented in Sect. 5. The experimental studies and the performance of our proposed approach in context of noisy speech is reported in Sect. 6. Finally, the paper concludes with discussions and directions for future research.

2 Baseline HMM Recognizer

The baseline recognizer in our study comprised of a standard HMM based isolated word recognizer trained using speech samples from TI46 database.

The database consisted of twenty isolated words (10 English words and 10 digits) from 8 males and 8 females totaling 160 examples for training and 256 examples for testing for each word class. Each word was modeled using a seven state HMM. Further, each HMM state was modeled using single Gaussian with diagonal covariance matrices. The baseline system was evaluated using HTK ToolKit v3.4 [10] and MFCC features computed with the following parameter settings: A hamming window of 25 ms, window shift of 10 ms, pre-emphasis coefficient set at 0.97, number of channels = 40 with energy and cepstral mean normalization options enabled. The 13 base coefficients were combined with delta and acceleration coefficients to derive a 39-dimensional final feature representation. Separate acoustic models were created for clean and noisy speech by varying the number of training examples and noise levels. The process is more fully described in Sect. 6. The next section discusses point-process representation based on spectral peaks.

3 Spectral Peak Based Point Process Representation of Speech

Speech signals in real world scenarios are often corrupted by noise that introduces distortions in the signal depending on noise type and strength. This often leads to degradation in recognition accuracy. The drop in recognition accuracy is non-linear and depends on the overall signal-to-noise ratio of the noisy speech signal. The relevant mitigation techniques can be applied at any of the following stages in the processing pipeline: during any pre-processing of the noisy speech signal or feature representation or modeling. In this work, we focus on a representation that will be robust to additive noise. The steps of obtaining such a representation from a speech utterance is described next.

3.1 Identification of Spectral Peaks

A linear speech spectrogram is obtained by applying short-term-Fourier-Transform (STFT) on speech signals of the spoken word template using overlapped Hamming windows at of length 25 ms. An overlap duration of 10 ms is maintained between successive frames. The window length is kept on the higher side to offer lower time resolution that results in the better continuity of the high energy formant tracks in the spectrogram image. The linear spectrogram is quantized into 40 frequency banks with mel-scale weighing followed by transformation to its log magnitude representation. The peaks in the log magnitude mel spectrogram are identified framewise and a matrix is created by assigning non-zero values (peak magnitudes) to the cells corresponding to the peak positions and zero otherwise. The matrix values are normalized between zero and one. Let $P = \{p_{f,b}\}$ be the set of spectral peaks obtained from log magnitude mel-scale spectrum after normalization where f and b represent the frame index and mel-bank index respectively. Let $V = \{v_{p_{f,b}}\}$ be the corresponding magnitudes. Here $0 \leq v_{p_{f,b}} \leq 1$, $f = \{1, 2, \dots, N\}$, $b = \{1, 2, \dots, M\}$, assuming N frames

and M mel-banks are used for analysis. Thus a spoken utterance is represented by its spectral peaks in a time-frequency matrix. On observation, it was found that under very low SNR conditions only the peaks having high magnitude were retained faithfully. Hence, it was decided to partition the peaks into three clusters depending on their normalized magnitudes and discard those with medium and low magnitudes. Let $C_h = \{h_{f,b}\}$ represent the cluster of peaks with high magnitudes and $V_h = \{v_{h_{f,b}}\}$ be the corresponding magnitudes. It is expected that the peaks in C_h , which are also referred to as the peaks will be affected least by noise. Thus the spectral peak based representation is a matrix P_D defined as:

$$P_D(f, b) = \begin{cases} v_{h_{f,b}} & \text{if } (f, b) \text{ is a spectral peak position} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

A visual inspection of the image of the matrix $P_D(f, b)$ reveals the presence of three distinct clusters of non-zero values. In the next step, the spectral peaks are further clustered and their values are assigned from the set $\{3^0, 3^1, 3^2\}$ depending on their membership in high, middle and low value clusters. This results in a matrix in which the spectral peaks correspond to the non-zero elements of the matrix. It is found that spurious spectral peaks (caused by noise) co-exist with genuine peaks particularly in the high frequency bands that needs to be eliminated along with empty frames at the word boundaries. The modeling is performed on the representation after removal of the spurious peaks.

3.2 Removal of Spurious Spectral Peaks

The steps for removal of such spurious peaks are described below:

1. Construct a new matrix M_D comprising of columns between the first non-zero columns from either ends of the matrix P_D .
2. Compute a matrix M'_D from M_D such that

$$M'_D(f, b) = \begin{cases} 1 & \text{if } M_D(f, b) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

3. Compute the standard deviation (σ_f) of positions of non-zero elements in each row of M'_D ,

$$\sigma_b = std(f) \text{ where } M'_D(f, b) \neq 0 \forall f \text{ and a given } b \quad (3)$$

4. Find

$$b' = \arg(\sigma_b > \tau), \tau \text{ being a word independent threshold} \quad (4)$$

5. Set $N_D(:, b') = M_D(:, b')$.
6. For each b' in N_D , let $r_b'^2$ and $r_b'^1$ respectively denote the continuous runs of 3^2 and 3^1 of length > 4 .

7. Set

$$N'_D(f, b'') = \begin{cases} 3^2 & \text{if } N_D(f, b') \in r_b^{f2} \\ 3^1 & \text{if } N_D(f, b') \in r_b^{f1} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

8. Set $M_D(:, b'') = N'_D(:, b'')$.

The matrix M_D denotes the spectral peak matrix after removal of empty boundary frames and spurious peaks. It is observed that rows containing spurious peaks have mostly non-zero elements spanning the entire row with few zeros in between. This causes the frame index values of the non-zero elements to have high variance. A word independent threshold τ determined empirically is kept on the variance for identification of such candidate rows. In our study, the value of τ has been set to 15. Figure 1 illustrates the steps for deriving spectral peak based representation for a clean instance of the spoken word *ZERO* and shows visual representation of each step of the process. It also illustrates the same for a noisy instance of the same word at *SNR* level of 10 dB.

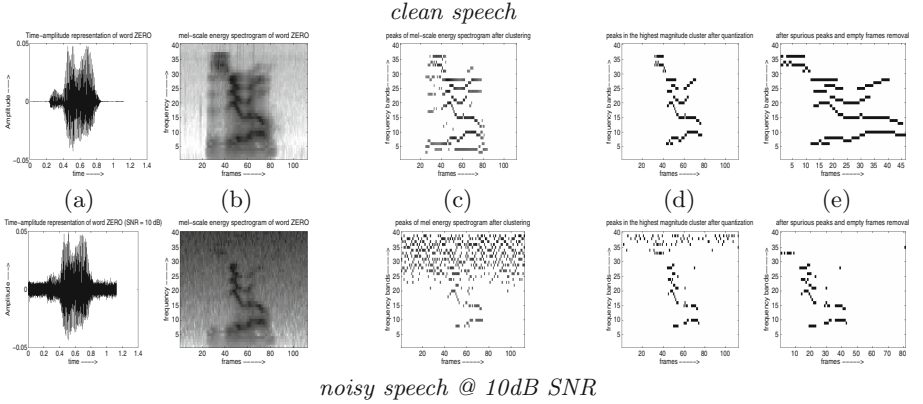


Fig. 1. Illustration of steps for representation based on spectral peaks. (a) speech waveform, (b) mel-scale weighted magnitude spectrogram, (c) clustering of peaks in the mel-scale magnitude spectrogram into three clusters, (d) peaks in the highest value cluster after assignment of quantized values and (e) representation after removal of spurious peaks in (d). The top and the bottom rows respectively illustrates the steps for clean and noisy (*SNR* = 10 dB) instances of spoken word *ZERO*.

4 Bankwise Segmentation of Spectral Peaks

An observation of the spectral peak representation reveals that the temporal variation of the peaks across different mel-bank frequency bins exhibits similarity within a given word class, however differing significantly across word classes. This implies that the representation can be used for characterizing spoken words.

It is seen that the dominant regions in the representation are limited to only certain concentrated regions in peak matrix. The group of banks that contribute to characterization of a particular word is estimated by bankwise profiling of peaks. Subsequently, point process models are derived for regions that exhibit concentrated presence of peaks. The steps for identification of such regions are as follows:

1. Let $\{M_{D_j}\}_{j=1}^{j=N}$ be the set of representations of N examples of spoken words of a given class.
2. Construct a matrix S by horizontal concatenation of all M_{D_j} 's.
3. Compute $r_S = \text{rowsum}(S)$. This results in a column vector having same number of rows as M_D .
4. Identify top B peaks (in terms of magnitude) in r_S and get the corresponding row indices for peak positions and width.
5. For each M_{D_j} , construct B sub-matrices comprising of rows identified in the previous step $\{M_{S_{ji}}\}_{j=1, i=1}^{j=N, i=B}$.

Each sub-matrix $M_{S_{ji}}$ represents a dominant region in the spectral peak representation matrix. Next, a bank specific amplitude detector is employed in each row of the sub matrix $M_{S_{ji}}$. The detector indicates the presence of a peak of a given amplitude (strength) in that bank with a binary indicator (0/1) per column. Thus independent detectors for detecting peaks of magnitude $\{3^0, 3^1, 3^2\}$ are employed in each row. The absence of a peak is indicated by a zero. This results in a new sparse matrix $M_{SS_{ji}}$ of dimension $3 \cdot m \times n$ where $M_{S_{ji}}$ has dimension of $m \times n$. The values of the elements of $M_{SS_{ji}}$ are assigned as follows:

$$\begin{bmatrix} M_{SS_{ji}}(3(x-1)+1, y), M_{SS_{ji}}(3(x-1)+2, y), \\ M_{SS_{ji}}(3x, y) \end{bmatrix} = \begin{cases} [0, 0, 0] & \text{if } M_{S_{ji}}(x, y) = 0 \\ [1, 0, 0] & \text{if } M_{S_{ji}}(x, y) = 3^0 \\ [0, 1, 0] & \text{if } M_{S_{ji}}(x, y) = 3^1 \\ [0, 0, 1] & \text{if } M_{S_{ji}}(x, y) = 3^2 \end{cases} \quad (6)$$

$M_{SS_{ji}}$ being a sparse matrix, motivates us to create a whole word model of a spoken word based on inhomogenous Poisson process modeling of spectral peak events.

5 Point Process Model Based Isolated Word Recognizer

Our small vocabulary isolated word recognizer consists of a set of bank specific amplitude detectors D_ϕ corresponding to a given bank amplitude combination ϕ as explained in Sect. 4. The set of bank amplitude combinations is denoted $\mathcal{F} = \{\phi_1, \phi_2, \dots, \phi_n\}$ where each $\phi \in \mathcal{F}$ corresponds to a row of the matrix $M_{SS_{ji}}$. The observations O are defined to be a collection of temporal point patterns $R = \{N_\phi\}_{\phi \in \mathcal{F}}$ generated by the detectors $\{D_\phi\}_{\phi \in \mathcal{F}}$.

The rate parameter $\lambda_\phi(t)$ is approximated by a piecewise continuous rate parameter that remains constant over a time segment. This is achieved by segmenting the word duration T into D segments, where the rate parameter of the d^{th} segment for a given ϕ is denoted by $\lambda_{\phi,d}$. The value of D is determined empirically and is set to 10 for reported experiments. The remaining formulation is same as in [2].

The values of the rate parameter $\lambda_{\phi,d}$ are derived independently for all ϕ 's and d 's using the sub-matrices $\{M_{SS_j i}\}_{j=1}^{j=N}$. Thus a model of each word class w is denoted by the set $\{pp_i^w\}_{i=1}^{i=B}$ each having B dominant regions and $pp_i^w = \{\lambda_{\phi,d}^w\}$ representing the point process model for the i^{th} region.

For a set of observations O from a test utterance, the recognition task can be formulated as finding the word class with highest posterior when tested against models of all word classes.

$$w^* = \arg \max_w \left[\sum_{i=1}^B P(O|pp_i^w) \right] \quad (7)$$

6 Experimentation and Results

The design of experiments was guided by two objectives. The first was to evaluate the robustness of our proposed approach on noisy speech with lesser number of training examples. The second was to compare its performance with HMMs. Towards this, a baseline HMM recognizer was trained on clean speech with MFCC features. The parameters for HMM modeling and MFCC feature extraction were mentioned in Sect. 2. Separate acoustic models were created by varying the number of training examples from 5 to 80. A PPM based recognizer was also trained in parallel on the same set of examples used for training HMMs following the process described in Sect. 5. This exercise was repeated three times, each for HMM and PPM by randomly choosing examples from the entire training set. Each run generated a set of models specific to the number of training examples used. This was to study the sensitivity of HMMs and PPMs to variation in input data. The reported recognition accuracies were obtained by averaging the recognition accuracy scores over three runs.

The testing was done both on clean and noisy utterances generated by adding white Gaussian noise to the clean speech samples at $\{0, 5, 10, 15\}$ dB SNR levels. The corresponding 1-best recognition accuracy figures (in %) for HMM with MFCC and PPM with spectral peak representation given in Table 1.

A comparative analysis of the two methods shows that under low resource conditions (10 training examples or less), our approach exhibits higher recognition performance compared to HMMs for all SNR levels. However, HMMs tend to outperform our proposed method when the number of training examples are increased. It is observed that the recognition accuracies for HMMs shows a sharp increase when the number of training examples crosses 20. It is also observed that for a given SNR condition, most of the learning in our approach saturates with few examples and the incremental increase in the recognition

Table 1. 1-best recognition performance of HMMs on noisy speech with models trained on clean speech using MFCC features. The number of training examples are varied during training. (maximum 80 examples per class).

Method	#train ex.	SNR (dB)				
MFCC+HMM		0	5	10	15	Clean
	5	5.14	11.12	25.37	36.57	72.34
	10	5.07	11.40	23.27	38.15	79.82
	20	13.1	28.81	50.81	62.05	93.95
	40	13.65	30.0	56.23	70.60	97.98
	80	19.27	41.95	60.90	75.30	99.80
Spectral peak+PPM	5	13.85	24.95	39.20	49.62	89.32
	10	13.93	25.06	39.85	49.8	89.79
	20	14.25	25.18	40.72	51.44	89.90
	40	14.27	25.21	40.85	51.62	90.03
	80	14.40	25.41	41.08	51.80	90.10

accuracy is insignificant with increase in the number of examples. This is probably because the positions and magnitude of the spectral peaks remains largely invariant across different instances of the same spoken word class resulting in no new learning with increase in the number of training examples. Also, the occurrences of such peaks are sparse (rare) events that are better modeled using Poisson process. The combined effect of the two results in better performance of our approach under low resource conditions. In addition, the study also investigates whether both the approaches can complement each other to explore the possibility of combining both towards improving the overall accuracy. Towards this, the confusion matrix derived from the outputs of HMM based recognizer corresponding to the highest achievable recognition scores at each SNR level is analyzed. The words that are misrecognized most (bottom five) by HMMs and their corresponding recognition accuracies are shown in Table 2. For these words, the recognition accuracies obtained by using PPM based recognizer is also reported.

It can be seen from Table 2 that for almost all words that are poorly recognized by the HMM based recognizer, the PPM based recognizer performs better often by several orders of magnitude. The only exception is the word *GO* whose short duration also results in poor recognition by PPMs, mostly attributed to weak duration modeling. This shows that HMM and PPM based recognizers can complement each other.

It is also to be noted that PPM of spectral peaks are achieving superior performance over HMMs under very low resource conditions without any assumption of any specific language. The pattern of spectral maxima events across different mel banks are only used to characterize the spoken words. The same can thus be potentially be applied for recognition of spoken words of any language.

Table 2. Words having the lowest (bottom five) 1-best recognition scores (in %) for HMM based recognition with 80 training examples per word class. The adjacent column reports the corresponding recognition scores for PPMs.

<i>SNR (dB)</i>	Method		
	Words	MFCC+HMM	Spectral peak+PPM
0	SEVEN	0.0	3.9
	START	0.0	12.7
	NINE	0.0	0.0
	ENTER	0.0	93.8
	REPEAT	0.0	31.5
5	SEVEN	0.0	12.5
	START	0.0	46.8
	NINE	2.4	13.5
	ENTER	3.9	32.0
	EIGHT	14.9	41.4
10	START	4.2	57.9
	NINE	8.5	33.3
	EIGHT	24.3	47.7
	GO	25.7	10.16
	ENTER	31.5	58.6
15	NINE	21.6	57.1
	START	25.1	65.0
	GO	28.2	14.0
	EIGHT	50.1	59.4
	STOP	65.1	66.4

7 Conclusion

In this work, a method has been proposed for robust recognition of noisy speech under low resource conditions. The proposed method uses keyword specific spectral peak events in selected groups of melbanks for obtaining a sparse representation of the speech. The temporal pattern of such peak events results in a family of independent point patterns that are modeled using an inhomogenous Poisson process model, generating whole word models of a spoken word. This method of spoken word representation and modeling shows an absolute improvement of in the range of 8–13%(absolute) over HMM systems using MFCC features when less than 10 training 10 examples are available for *SNR* levels from 0–15 dB. Greater improvements in recognition accuracies are observed for speech at higher *SNRs* under similar constraints of training examples. In case of similar recognition accuracy figures for a given *SNR* level, our proposed approach requires much lesser number of training examples compared to HMM. Analysis of confusion

matrices reveal that recognizers trained using HMMs on MFCC features and PPMs on spectral peaks show complementary results. This was inferred from the observation that the later approach had exhibited better recognition performance on those examples that were poorly recognized by HMMs. Future work can hence be undertaken towards combining both the approaches towards a higher overall recognition score. It might also be interesting to study the effect of considering spectral peaks from all mel banks instead of a selected group of mel banks as a future work.

References

1. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Sig. Process. Mag.* **29**(6), 82–97 (2012)
2. Jansen, A., Niyogi, P.: Point process models for spotting keywords in continuous speech. *IEEE Trans. Audio Speech Lang. Process.* **17**(8), 1457–1470 (2009)
3. Jansen, A., Dupoux, E., Goldwater, S., Johnson, M., Khudanpur, S., Church, K., Feldman, N., Hermansky, H., Metze, F., Rose, R., et al.: A summary of the 2012 JHU CLSP workshop on zero resource speech technologies and models of early language acquisition (2013)
4. Jansen, A., Mesgarani, N., Niyogi, P.: Point process models of spectro-temporal modulation events for speech recognition. In: 2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), pp. 104–108. IEEE (2010)
5. Jansen, A., Niyogi, P.: Robust keyword spotting with rapidly adapting point process models. In: Tenth Annual Conference of the International Speech Communication Association (2009)
6. Jansen, A., Niyogi, P.: Detection-based speech recognition with sparse point process models. In: 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 4362–4365. IEEE (2010)
7. Liu, C., Jansen, A., Khudanpur, S.: Context-dependent point process models for keyword search and detection-based ASR. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6025–6029. IEEE (2016)
8. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **77**(2), 257–286 (1989)
9. Wang, Y., Yang, J.A., Lu, J., Liu, H., Wang, L.W.: Hierarchical deep belief networks based point process model for keywords spotting in continuous speech. *Int. J. Commun. Syst.* **28**(3), 483–496 (2015)
10. www.htk.eng.cam.ac.uk/docs/docs.shtml
11. www.ldc.upenn.edu/Catalog/ti46.readme.html

Significance of DNN-AM for Multimodal Sentiment Analysis

Harika Abburi¹(✉), Rajendra Prasath², Manish Shrivastava¹,
and Suryakanth V. Gangashetty¹

¹ Language Technology Research Center, International Institute of Information
Technology Hyderabad, Hyderabad, India

harika.abburi@research.iiit.ac.in, {m.shrivastava,svg}@iiit.ac.in

² IIIT, Sricity, India

drprasad@gmail.com

Abstract. The furtherance of social media led people to share the reviews in various ways such as video, audio and text. Recently, the performance of sentiment classification is achieved success using neural networks. In this paper, neural network approach is presented to detect the sentiment from audio and text models. For audio, features like Mel Frequency Cepstral Coefficients (MFCC) are used to build Deep Neural Network (DNN) and Deep Neural Network Attention Mechanism (DNNAM) classifiers. From the results, it is noticed that DNNAM gives better results compared to DNN because the DNN is a frame based one where as the DNNAM is an utterance level classification thereby efficiently use the context. Additionally, textual features are extracted from the transcript of the audio input using Word2vec model. Support Vector Machine (SVM) and Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) classifiers are used to develop a sentiment model. From the experiments it is noticed the LSTM-RNN outperforms the SVM as the LSTM-RNN is able to memorize long temporal context. The performance is also significantly improved by combining both the audio and text modalities.

Keywords: Multimodal sentiment analysis · MFCC · Word2vec
SVM · Deep neural networks · DNNAM · LSTM-RNN

1 Introduction

With rise of social media, most of the information available in the Internet is in the form of video, audio, image and text. As each modality have some special information in it, all kinds of information are required for classification of the input. For any kind of approach like audio, video and text, sentiment can be extracted using machine learning approaches [22]. The sentiment analysis plays a key role in classification of many applications such as in online songs [3,9], reviews [16,23] and twitter information [1,2].

Speech data is extracted from the vocal track, prosody and excitation. The prosodic and spectral features can be used to build the sentiment classifier [6]. Audio features like pitch, intensity and loudness, are extracted using OpenEAR toolkit and SVM, Hidden Markov Models (HMM), Gaussian Mixture Model (GMM) classifiers are built to identify the sentiment [13,21]. Instead of using SVM, HMM and GMM, currently research work is going on the deep neural networks which has ability to discover multiple levels of features from input. Deep neural networks (DNN) are applied in many applications like speech, language recognition, sentiment analysis, etc. By using deep neural networks in acoustic modeling for speech recognition the performance is significantly improved [8]. In [7,11], a single DNN acoustic model is used to train both the language recognition and speaker recognition tasks. For sentiment classification also DNN model is built and observed significant improvement compared to GMM [10]. The drawback of the DNN system is that, the decision is taken at every frame and the context used is fixed which is assigned to the entire utterance and it cannot memorize long temporal context. To overcome this problem, a feed-forward deep neural network with an attention mechanism [5] is proposed to solve long range dependency memory problems. This attention mechanism is used for the language recognition task in [14], which results in better performance compared to the DNN. In this paper, we used this architecture for sentiment analysis and referred as deep neural network attention mechanism (DNNAM). This attention mechanism is parallelized because it is a feedforward neural network with no recurrent networks and it is able to memorize the long temporal context. This architecture is used for classifying whole utterance where as in DNN a frame-level decision is considered and all the decisions are combined to extract the sentiment.

For text sentiment classification a variety of features like unigrams, bigrams and combination of both are explored, and SVM classifier is built using these features. To detect the polarity of tweets two different Naive Bayes classifiers are developed using the polarity lexicon [15]. The most commonly used text based classification techniques like SVM, Maximum Entropy and Naive Bayes are based on a bag of words model in which the sequence of words is ignored. This may result in inefficient in extracting the sentiment from the input because the sequence of words will affect the sentiment present in it. By overcoming this problem many researches reported by employing deep learning in sentiment analysis. A deep neural network architecture that jointly uses sentence, word, character levels to perform sentiment analysis is proposed [4]. A DNN is applied for language modeling [24] which outperforms the n-gram model. A DNN can have any number of hidden layers and any number of nodes in each layer in which the weights are connected in between. A DNN can learn a more complex model when the layers of architecture are more. But, a simple feedforward neural network cannot be more accurate just by only adding layers because the training process is ineffective when there are more layers [12] and it may not capture the temporal context accurately. To capture the temporal context better Recurrent Neural Network (RNN) has been proposed [19]. They applied RNN on speech

recognition for language modeling. It has been shown that RNN outperforms n-gram technique. The advantage of RNN is that it will use previous state information to compute its current state, which is similar to the context in most of the natural languages. However, simple RNN has a problem in passing the long sequence. A solution to this is LSTM, a RNN with additional long term memory, that was proposed in [17]. In this paper an LSTM-RNN classifier is proposed to extract the sentiment from the input.

Multimodal sentiment analysis is a combination of more than one modality to identify the sentiment present in the data. Modalities can be combined by using feature or decision level fusions. By combining different modalities, an improvement over classification is observed when compared to a single modality [18]. A Spanish database [20] is collected and explored the combination of three modalities such as audio, video and text features which results in significant improvement compared to single modality. In this work, audio and text modalities are combined to detect the sentiment from the reviews database. From the literature, most of the sentiment models are built using the several features which are extracted using OPENEAR/OPENSIMILE tools. Instead of using tools, MFCC features are used to build the sentiment models using DNNAM classifier. To detect the sentiment from the text, features which are computed using word2vec are used to build the LSTM-RNN classifier. Finally, sentiment is extracted in combination of both the modalities.

Rest of the paper is organized as follows: Databases used are discussed in Sect. 2. Audio sentiment classification is proposed in Sect. 3. Text sentiment classification is described in Sect. 4. Multimodal sentiment analysis and experimental results of proposed method is presented in Sect. 5. Finally, Sect. 6 concludes this study with a mention on the future scope of this work.

2 Databases

2.1 Spanish Database

This database used in this study is MOUD (Multimodal Opinion Utterances Dataset) which is obtained from [20]. Database is collected from YouTube, consists of a variety of product reviews on book, movies and perfume. The dataset has 100 videos, among them 42 positive, 22 neutral and 36 negative. Two basic sentiments considered are negative and positive. Among them 80% is used for training and 20% is used for testing. For text based sentiment classification, sentiment annotations and transcription are done manually. The average number of words in each input is around 50. Annotators were provided with both the audio and transcribed text to correctly figure out the opinion.

2.2 Hindi Database

The database is obtained from [10]. The dataset has reviews on various phones, shampoos and lotions. As variety of product reviews are used, the database

has some degree of generality. Both the modalities such as audio and text are provided for annotators to figure out the exact opinion of the input. A total of 110 reviews are collected, among them based on inter-annotator agreement 100 reviews are considered. The average number of words in each input is around 40 and the average length of each input is thirty seconds. Transcriptions were manually performed for text sentiment classification.

3 Audio Sentiment Classification

The classifiers like Gaussian Mixture Models (GMM), DNN and DNNAM are built using MFCC features. Recently sentiment analysis model is build using the DNN classifier [10] and achieved good performance. A DNN can have any number of hidden layers and any number of nodes in each layer. But, a simple feedforward neural network or DNN cannot achieve good performance just by adding more layers to it because the training process is inefficient if there are more layers. The main drawback of the DNN system is that, at each frame decision is taken and the context used is fixed which is usually assigned to an entire utterance. DNN cannot memorize the long temporal context. To overcome these problems a DNNAM architecture is proposed to better capture the temporal context and to do utterance-wise sentiment classification.

3.1 DNNAM

A DNNAM is a simple DNN implement with attention mechanism. The advantage of DNNAM is, it is able to memorize and is also parallelized because of the strictly feedforward neural network. Rather than taking a frame level decision as done in DNN here an entire utterance is classified. The attention mechanism will also go deep into the input feature frames which are more important to extract the sentiment. In the DNNAM architecture, the attention is computed by using the input feature vectors. The block diagram for deep neural network attention model is shown in Fig. 1.

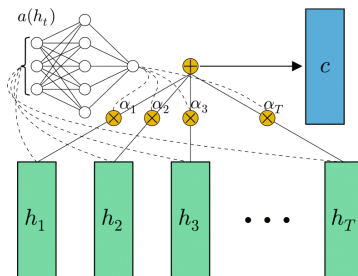


Fig. 1. Deep neural network attention model [5]

Given an input sequence, $Y = \{Y_1, Y_2, Y_3, \dots, Y_T\}$, a hidden layer sequence $H = \{h_1, h_2, h_3, \dots, h_T\}$, is computed by the DNN and attention is computed on the hidden features.

The attention mechanism $a(h_t)$ which is shown in Fig. 1 is computed using single layer perceptron and then softmax operation is performed to normalize the values between zero and one.

$$H = [h_1 \ h_2 \ h_3 \ \dots \ h_T] \quad (1)$$

$$\gamma = \tanh(W_a H + b_a) \quad (2)$$

$$\alpha = \text{softmax}(\gamma) \quad (3)$$

In the above equations, γ is a hyperbolic tanh function and W_a, b_a are the parameters of the attention mechanism. By using back propagation algorithm these parameters are optimized, where as α is referred to a attention vector. The hidden state sequence vectors h_t are fed into the attention mechanism $a(h_t)$ to produce a probability vector α .

The context vector c which is shown in Fig. 1 is computed using weighted average of H with weight α .

$$c = H\alpha \quad (4)$$

Then, the output vector is computed by transforming the c using output layer weights U and then softmax operation is performed.

$$Z = \text{softmax}(Uc + b_o) \quad (5)$$

In the above equation b_o is the output bias. From the above equation it is infer that from the entire input utterance Y , a single decision vector Z is predicted.

The architecture can have any number of layers before or after the attention model. The number of layers in this architecture are termed as the hidden layers present before the context vector plus one additional output layer. This architecture is trained for different number of epochs and with the ADAM method which is hyper parameter learning algorithm. MFCC features considered are 13-dimensional, 65-dimensional (five successive MFCC frames are combined) and 130-dimensional (ten successive MFCC frames are combined) feature vectors. As single frames cannot carry the sentiment, the frames are combined and observed that combination of frames results in significant improvement in performance compared to each frame. The input layer units are linear that can have 13 or 130 or 65 nodes and the output layer is of softmax layer with two nodes as two classes are considered for this study. For testing five seconds of data is used and the node which gives maximum score is assigned as the claimed class.

From the Tables 1 and 2 it is observed that DNNAM performance is better compared with DNN which is proposed in [10] because DNNAM with hidden layers before the attention mechanism captures the context and put the information before getting the final decision. From the Table 2 it is also observed that DNNAM with 65-Dimensional feature vector has performed better compared to 130-Dimensions because when we combine frames, features are less therefore they are not enough to train the DNNAM.

Table 1. Performance (in %) of sentiment analysis using deep neural network for Hindi and Spanish databases.

Features	Hindi [10]		Spanish	
	2 layers	3 layers	2 layers	3 layers
13-Dimensional	41.6	58.3	50.0	61.2
65-Dimensional	58.3	75.0	66.7	72.0
130-Dimensional	58.3	66.7	61.2	66.7

Table 2. Performance (in %) of sentiment analysis using deep neural network attention mechanism.

Features	Hindi		Spanish	
	2 layers	3 layers	2 layers	3 layers
13-Dimensional	50.0	62.5	56.2	68.7
65-Dimensional	68.7	81.2	72.0	77.4
130-Dimensional	62.5	68.7	61.2	75.0

Table 3. Performance (in %) of sentiment analysis using different classifiers.

Classifiers	Hindi	Spanish
SVM [20]	-	46.7
GMM	58.3	66.4
DNN	75.0 [10]	72.0
DNNAM	81.2	77.4

Experiments are also done using GMM classifier. GMM is tested with different number of mixtures like 16, 32 and 64. From the Table 3 it is observed that DNNAM outperforms the DNN [10] and GMM with 64 mixtures for the databases. It is also observed that the performance of spanish data with DNNAM classifier is improved significantly [20].

4 Sentiment Analysis Using Text Features

Word2Vec is used to represent the input in fixed length feature vector. This feature vector is used as input for the classifiers like SVM and LSTM-RNN.

LSTM-RNN and SVM are build for text classification. By using Word2Vec vectors, SVM classifier is trained. Word2Vec takes data from a corpus, and churns out vectors for each word. The length of word vector is independent from the size of dictionary. This vector will be input to a machine learning algorithms. The problem with SVM is that it just uses the given features directly without modeling the sequence information. To overcome this, experiments performed on LSTM-RNN model.

A recurrent neural network (RNN) is a network of neurons with feedback connections. It can learn sequence tasks that are not possible by the traditional methods such as feedforward networks and SVM which does not have any internal states at all. In spite of its advantages RNN suffers from vanishing gradient descent problem which is overcome by LSTM-RNN. LSTM-RNN prevent the back propagated errors from vanishing. The errors can flow backwards through any number of virtual layers which are unfolded in space. The LSTM-RNN model will automatically learn a flexible history length which has an abstracted feature representation. The LSTM-RNN model [17] is a recurrent neural network model with multiple hidden layers and a special memory unit. In LSTM-RNN, the information is stored in two ways: Long-term Memory as weights and Short-term Memory. LSTM-RNN can capture the long dependencies in a sequence by introducing a memory unit and a gate mechanism which aims to decide how to utilize and update the information kept in the memory cell.

In this work an LSTM network with 32 units and a single output neuron with a softmax activation function is used for making 0 or 1 predictions for the two classes. A log loss function is used and the network is optimized using the ADAM optimization function. The model is fit over 50 epochs with a batch size of 128.

Table 4. Performance (in %) of sentiment analysis using text features for both the databases.

Classifier	Hindi	Spanish
SVM	65.5	63.6
LSTM-RNN	72.4	68.3

From Table 4, it is noticed that the performance for Hindi and Spanish databases is high with LSTM-RNN compared to SVM.

5 Multimodal Sentiment Analysis

As sentiment classification with both textual and audio features had few limitations, in our work multimodality sentiment analysis is implemented. A decision level fusion is implemented here, means after calculating average probabilities for both modalities which class gives highest average probability that test case is hypothesized from that class. Compared to textual data, audio information provide more natural experience as it allows the listener to better sense the human being intentions. From experimental results, it is observed that the combination of two modalities create a better sentiment analysis model compared to individual modality.

From the Table 5, it is observed that by combining both the two modalities, rate of detecting the sentiment is improved significantly compared to individual modality.

Table 5. Performance (in %) of multimodal sentiment analysis for Hindi and Spanish datasets.

Modality	Hindi	Spanish
Audio	81.2	77.4
Text	72.4	68.3
Audio + Text	85.6	82.3

6 Summary and Conclusions

In this paper, an approach to extract sentiment from audio and text modalities using deep neural networks is presented. For audio, DNNAM, DNN and GMM classifiers are built using the MFCC features. From our experimental results, it is observed that DNNAM classifier outperformed DNN and GMM classifiers as the attention mechanism captures the context and put the information before taking the decision for an utterance. DNNAM classifier with 65-dimensional MFCC features are more accurate compared to 130-dimensional because when we combine frames number of features are less therefore not enough to train the classifier. For text, Word2Vec vectors are used to build the classifiers such as SVM and LSTM-RNN. From experimental results, it is observed that LSTM-RNN outperforms the SVM classifier because SVM does not model sequence information. The performance of sentiment is significantly improved by combining both the modalities.

References

1. Kumar, A., Sebastian, T.M.: Sentiment analysis on Twitter. *Int. J. Comput. Sci. (IJCSI)* **9**(4), 372–378 (2012)
2. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of Twitter data. In: *Proceedings of Workshop on Languages in Social Media*, pp. 30–38 (2011)
3. Patra, B.G., Das, D., Bandyopadhyay, S.: Mood classification of Hindi songs based on lyrics. In: *Proceedings of 12th International Conference on Natural Language Processing (ICON)* (2015)
4. dos Santos, C.N., Gatti, M.: Deep convolutional neural networks for sentiment analysis of short texts. In: *Proceedings of 25th International Conference on Computational Linguistics (COLING)*, pp. 69–78 (2014)
5. Raffel, C., Ellis, D.P.W.: Feed-forward networks with attention can solve some long-term memory problems. In: *CoRR*, vol. abs/1512.08756 (2015). <http://arxiv.org/abs/1512.08756>
6. Mairesse, F., Polifroni, J., Di Fabbrizio, G.: Can prosody inform sentiment analysis? Experiments on short spoken reviews. In: *Proceedings of IEEE International Conference on Acoustics, Speech, Signal processing (ICASSP)*, pp. 5093–5096 (2012)
7. Richardson, F., Reynolds, D., Dehak, N.: A unified deep neural network for speaker, language recognition. In: *Proceedings of INTERSPEECH*, pp. 1146–1150 (2015)

8. Hinton, G., Deng, L., Dong, Y., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., Kingsbury, B.: Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Process. Mag.* **29**(6), 82–97 (2012)
9. Abburi, H., Akkireddy, E.S.A., Gangashetty, S.V., Mamidi, R.: Multimodal sentiment analysis of Telugu songs. In: *Proceedings of 4th Workshop on Sentiment Analysis where AI meets Psychology (SAAIP)*, pp. 48–52 (2016)
10. Abburi, H., Prasath, R., Shrivastava, M., Gangashetty, S.V.: Multimodal sentiment analysis using deep neural networks. In: Prasath, R., Gelbukh, A. (eds.) *MIKE 2016. LNCS (LNAI)*, vol. 10089, pp. 58–65. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58130-9_6
11. Lopez-Moreno, I., Gonzalez-Dominguez, J., Plchot, O., Martinez, D., Gonzalez-Rodriguez, J., Moreno, P.: Automatic language identification using deep neural networks. In: *Proceedings of IEEE International Conference on Acoustic, Speech, Signal Processing (ICASSP)*, pp. 5337–5341 (2014)
12. Deng, L.: A tutorial survey of architectures, algorithms, applications for deep learning. *APSIPA Trans. Signal Inf. Process.* **3**, 1–29 (2014)
13. Morency, L.P., Mihalcea, R., Doshi, P.: Towards multimodal sentiment analysis: harvesting opinions from the web. In: *Proceedings of 13th International Conference on Multimodal Interfaces (ICMI)*, pp. 169–176, November 2011
14. Mounika, K.V., Sivanand, A., Lakshmi, H.R., Gangashetty, S.V., Vuppala, A.K.: An investigation of deep neural network architectures for language recognition in Indian languages. In: *Proceedings of INTERSPEECH*, pp. 2930–2933 (2016)
15. Gamallo, P., Garcia, M.: Citiuis: a naive-bayes strategy for sentiment analysis on English Tweets. In: *Proceedings of 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pp. 171–175, August 2014
16. Singh, R., Kaur, R.: Sentiment analysis on social media, online review. *Int. J. Comput. Appl.* **121**(20), 44–48 (2015)
17. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
18. Poria, S., Cambria, E., Howard, N., Huang, G.-B., Hussain, A.: Fusing audio, visual, textual clues for sentiment analysis from multimodal content. *Neurocomputing* **174**, 50–59 (2015)
19. Mikolov, T., Karafiat, M., Burget, L., Cernocky, J.H., Khudanpur, S.: Recurrent neural network based language model. In: *Proceedings of INTERSPEECH*, pp. 1045–1048 (2010)
20. Perez-Rosas, V., Mihalcea, R., Morency, L.-P.: Multimodal sentiment analysis of Spanish online videos. *IEEE Intell. Syst.* **28**(3), 38–45 (2013)
21. Perez-Rosas, V., Mihalcea, R., Morency, L.-P.: Utterance level multimodal sentiment analysis. In: *Proceedings of ACL*, pp. 973–982 (2013)
22. Medhat, W., Hassan, A., Korashy, H.: Sentiment analysis algorithms, applications: a survey. *Ain Shams Eng. J.* **5**(4), 1093–1113 (2014)
23. Fang, X., Zhan, J.: Sentiment analysis using product review data. *J. Big Data* **2**(5), 1–14 (2015). Springer Open Journal
24. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. *J. Mach. Learn. Res.* **3**, 1137–1155 (2003)

Pattern Based Information Retrieval Approach to Discover Extremist Information on the Internet

Mikhail Petrovskiy, Dmitry Tsarev^(✉), and Irina Pospelova

Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University,
Moscow, Russia

{michael,tsarev}@cs.msu.su, ipospelova05@yandex.ru

Abstract. This paper is devoted to the research and development of machine learning methods aimed at discovering potentially dangerous extremist information in social networks using pattern based approach. In this approach, a text document containing extremist information is used for automatic extracting keywords to query a social networks search engine, and then found messages are filtered according to the topic based measure of relevance with the pattern. N-gram based algorithms are proposed for constructing hidden topics and keywords that allow applying the proposed approach in the case of multilingual and illiterate texts. The performance of the proposed methods is experimentally studied on benchmark AnsarI dataset.

Keywords: Social network analysis · Anti-extremist intelligence
Latent semantic analysis · Keywords extraction · Orthogonal NMF · N-gram

1 Introduction

During the last decade, terrorist and extremist organizations have significantly increased their presence on the Internet and in social media. These tools are used for recruiting and training new members, for preparing and organizing terrorist attacks, for promoting violence, distributing extremist literature, etc. The Internet is global, free, and open resource. It allows disseminating any information fast and anonymously, addressing directly to the audience of social networks and forums, without fear of censorship existing in traditional mass media. Security measures such as identifying terrorists and persons associated with them, stopping the distribution of extremist literature, preventing planned terrorist attacks require analysis of all information received from the members of potentially extremist groups. Therefore, the analysis of the Internet resources comes to the fore. The aim of such analysis is identifying potentially dangerous users, prompt removing extremist materials, analyzing information on terrorists and upcoming terrorist incidents. Large amount of information, published on various languages, distributed through the Internet should be monitored in real-time. Therefore, it is necessary to use automated text analysis procedures.

This paper presents an approach for extremist information detecting on the Internet. The approach is based on the latent semantic analysis (LSA) [1] applied to a text with extremist information for finding specific hidden topics and keywords. Keywords are

used to form search queries for popular social networks (such as twitter, vk, etc.). All found documents and messages are ranked according to their relation to the extracted topics using specially developed topic based relevance measure.

In the case of detecting potentially extremist information in social networks, the analysis should take into account the following features:

- different parts of a single text may be written on different languages (e.g., Russian, English and Arabic in one document);
- texts may contain slang and lingo words used only in closed communities and unknown to others;
- texts may suffer from grammar errors and usage of special coded words (such as coded names of drugs, weapons, particular persons and locations).

These peculiarities lead to the requirement of language independent preprocessing of words in a document. We satisfy this requirement using N-grams [2] for words representation and applying LSA (topic modeling) based on orthogonal nonnegative matrix factorization [3, 4] for N-gram representation of the text to discover main topics. Then the topics are used for finding language independent keywords for querying search engine. Finally, the relevance measure based on the found topics is being applied to find semantically close documents in the search engine output.

This paper is organized as follows. The suggested approach is described in Sect. 2. Experimental evaluation on benchmark data is presented in Sect. 3. Finally, conclusions are formulated in Sect. 4.

2 Proposed Approach

2.1 General Scheme

The suggested approach is represented as a three-stage scheme (see Fig. 1).

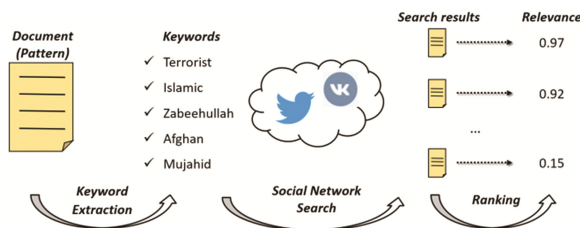


Fig. 1. General scheme of the proposed approach.

Stage 1 extracts keywords from a pattern document; stage 2 uses the extracted keywords to run search queries in social networks and saves the documents found; stage 3 ranks the found documents according to their relation to the main topics of the pattern document. On stages 1 and 3, the topic models are used. The topic modeling is based on orthogonal nonnegative matrix factorization (ONMF) of a pattern document. Orthogonality of NMF is necessary, because one needs to apply the obtained topic model to

new documents, i.e. to projects the found documents to the topic space of the initial pattern document. The projection provides the possibility of estimating their relevance to the initial document. Usually latent semantic analysis uses text data represented as bag-of-words. In this model, lexemes of the text are used as text's features (terms). In our approach, lexemes extracted from the text are presented in the form of N-grams. An N-gram is an N-character slice of a longer string (for example, lexeme "TEXT" is represented in the form of 3-grams as the sequence: _TE, TEX, EXT, XT_) [2]. It makes possible to use the model for multilingual and poor quality texts. The following points are considered below in this section:

- matrix representation of the text from the pattern document;
- construction of a topic model for the pattern document with the use of ONMF;
- keywords extraction from the text using topics representation and run search query;
- relevance estimation of the documents found in social networks on the basis of the topic model of the pattern document.

2.2 "N-Gram Vs Sentence" Matrix Text Representation Model

We use the modified documents vector model, which presents a text of the pattern document as a numeric matrix $A \in \mathbb{R}^{m \times n}$ with rows corresponding to N-grams and columns corresponding to the sentences of the text, not to the whole text as in traditional approach. The sentence A_j ($1 \leq j \leq n$) is represented as a numeric vector $A_j = [a_{1,j}, a_{2,j}, \dots, a_{m,j}]^T$ of a fixed length m , where m is the dimension of N-gram vector space and $a_{i,j}$ is i th ($1 \leq i \leq m$) component of the vector A_j that stands for the weight of i th N-gram in j th sentence. Weight $a_{i,j}$ is calculated as product of three factors: $a_{i,j} = L_{i,j} \cdot G_i \cdot N_j$, where $L_{i,j}$ is local weight of feature i in the sentence j , G_i is global weight of feature i in all sentences, N_j is normalization of the vector A_j [5, 6]. Combination of local weight, global weight and normalization is called a *weight scheme*. The weight scheme is chosen depending on the problem to be solved.

On the keywords extraction stage, we use the following scheme: *binary* local weights, where $L_{i,j} = 1$ if i th N-gram presents in j th sentence and $L_{i,j} = 0$ otherwise, *entropy*-based global weights

$$G_i = 1 - \sum_{j=1}^N \left(\frac{p_{i,j} \log p_{i,j}}{\log N} \right), p_{i,j} = \frac{t_{i,j}}{F_i}, F_i = \sum_{k=1}^N t_{i,k}, \quad (1)$$

where $t_{i,j}$ is a number of occurrence of i th N-gram in j th sentence, and *no normalization* with $N_j = 1$.

For relevance estimation we use another scheme: *logarithmically scaled frequency* local weights

$$L_{i,j} = 1 + \log(t_{i,j}), \quad (2)$$

where $t_{i,j}$ is a number of occurrence of i th N-gram in j th sentence, *IDF* global weighs

$$G_i = 1 + \log(N/n_i), \tag{3}$$

where N is the number of sentences in the document and n_i is the number of sentences containing i th N-gram, and cosine normalization

$$N_j = 1 / \sqrt{\sum_{i=0}^m (L_{i,j} G_i)^2}. \tag{4}$$

Both schemes were chosen experimentally.

The usage of the N-gram approach does not require any additional complex linguistic preprocessing of a text. Dividing words to N-grams is much easier than stemming. Each language has finite alphabet, therefore maximum number of different features (N-grams) is also finite. The main disadvantage of the method is that N-grams increases number of features very much in comparison with usual term-based approach, especially for small n . Besides, usage of N-grams leads to non-interpreted text model. Bag-of-words model does not take into account the order of the features in the text. Therefore, the text composed of words divided to N-grams is not understandable as a sequence of words. Moreover, analyzing text as N-grams matrix, it is difficult to find out words presented in the initial text. This problem arises in keywords extraction.

2.3 Extracting Hidden Topics Using ONMF

In our approach, ONMF is used for topics extraction. Being applied to the text presented in the form of matrix $A \in \mathbb{R}^{m \times n}$, ONMF finds a mapping matrix of the space of k topics to the space of m N-grams $W_k \in \mathbb{R}^{m \times k}$ and a matrix of representation of the text sentences in the topic spaces $H_k = [H^1, \dots, H^n] \in \mathbb{R}^{k \times n}$. ONMF factorizes the matrix $A \in \mathbb{R}^{m \times n}$ into nonnegative matrices $W_k \in \mathbb{R}^{m \times k}$ and $H_k \in \mathbb{R}^{k \times n}$ that minimize the following objective [4]:

$$f(W_k, H_k) = \frac{1}{2} \|A - W_k H_k\|_F^2 + \frac{\alpha}{2} \|W_k^T W_k - I\|_F^2, \tag{5}$$

where $k \ll \min(m, n)$ and $\alpha \geq 0$ is regularization parameter needed to ensure the matrix W_k is close to its orthogonal and normalized form. That is, if $\alpha > 0$ then additional condition is to be satisfied: $W_k^T W_k = I$. Matrices $W_k \in \mathbb{R}^{m \times k}$ and $H_k \in \mathbb{R}^{k \times n}$ are obtained as the result of ONMF applied to the text matrix $A \in \mathbb{R}^{m \times n}$, they have the following properties (see Fig. 2).

Columns of the matrix W_k correspond to k topics extracted from the text. Element w_{ij} defines weight of the i th N-gram for the j th topic. The higher is the value of w_{ij} in comparison with other elements of the j th column (j th topic), the more typical is i th N-gram for the given topic. Therefore, the extracted topics may be described by the N-grams with the highest weight. Matrix $W_k^T \in \mathbb{R}^{k \times m}$ maps the space of m N-grams to the space of k topics. Nonnegative element of the matrix H_k may be considered as a contribution (weight) of the topic to the corresponding sentence. The higher is the value of an

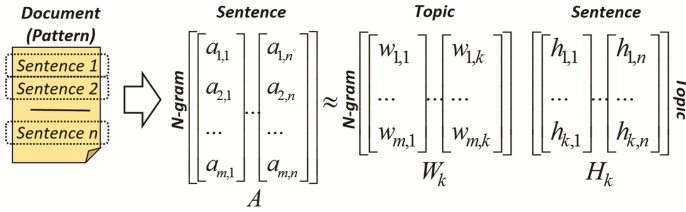


Fig. 2. ONMF of a pattern document.

element h_{ij} compared with the other elements of j th column, the more is the meaning of i th topic to the sentence. This property is widely used in text data clusterization [3, 7].

The solution of (5) for fixed α can be found by multiplicative update rules based algorithm [4] (see Fig. 3). The update rules are obtained from the Karush–Kuhn–Tucker (KKT) conditions for the objective (5).

1. All elements of matrices $W^1 \in \mathbb{R}_+^{m \times k}$ and $H^1 \in \mathbb{R}_+^{k \times n}$ are initialized by random positive values
2. Run loop by $p = 1, \dots, Q$ to calculate W and H :
 - a. $H_{b,j}^{p+1} = H_{b,j}^p \cdot \frac{((W^p)^T \cdot A)_{b,j}}{(W^p)^T \cdot W^p \cdot H^p}_{b,j}, \forall b, j: 1 \leq b \leq k, 1 \leq j \leq n$
 - b. $W_{i,a}^{p+1} = W_{i,a}^p \cdot \frac{(A \cdot (H^{p+1})^T + \alpha \cdot W^p)_{i,a}}{(W^p \cdot H^{p+1} (H^{p+1})^T + \alpha \cdot W^p \cdot (W^p)^T \cdot W^p)_{i,a}},$
 $\forall i, a: 1 \leq i \leq m, 1 \leq a \leq k$

Fig. 3. Iterative ONMF algorithm.

2.4 Using N-Gram Based Topics for Keywords Extraction and Document’s Relevance Estimation

In usual case, interpretability of the topic space constructed using ONMF allows to describe each topic by a set of key terms. This useful feature is absent in our case, when we use N-grams instead of terms. Obviously, a set of key N-grams is not informative for human understanding. To discover keywords, we map back N-gram based topics to the space of terms. Each word $Z_i (1 \leq i \leq q, q$ is number of different words of the text)

of the analyzing document is represented as the vector $Z_i = [z_{1,i}, z_{2,i}, \dots, z_{m,i}]^T$ in the N-gram space of the pattern document. Thus, we have a matrix $Z \in \mathbb{R}^{m \times q}$ of all words of the text. Matrix Z is mapped to the obtained topic space by multiplying by W_k^T . The resulting matrix

$$Hz = W_k^T Z \tag{6}$$

corresponds to representation of all words of the text in the topic space. Element $H z_{i,j}$ shows the correspondence between j th word and i th topic. To compose a set of keywords

for each topic i ($1 \leq i \leq k$), we choose p words with indices (j_1, j_2, \dots, j_p) corresponding to maximum elements in i th row of the matrix H_z . Normalization of a word's weight by its length in symbols $H_{z_{i,j}} = H_{z_{i,j}} / (1 + \text{len}(Z_j))$ is well established. This approach gives greater weights to short words. The greatest weight has the shortest of all morphological forms of a word.

Keywords extracted by the proposed method are used for querying social network or web search engine to obtain the list of documents (web resources, messages, forum threads, etc.) containing these keywords. In the case of detecting extremist information, the most part of the search engine output obtained with the extracted keywords usually consists of irrelevant document such as news articles, religious (not terrorist) discussions and poetry, etc. That is why the next step is filtering or ranking found documents according to their relevance to the initial pattern documents with extremist information. In our approach, the relevance measure of a new found document is based on its representation in the topic space of the pattern document. Let's assume that the representation of the new document in "N-gram vs sentence" space is described by matrix B . To map the document to the space of k topics, one needs to multiply the matrix B by the matrix W_k^T . The resulting matrix

$$H_{new} = W_k^T B \tag{7}$$

corresponds to the representation of the new document in the form of "N-gram vs topic" space of the pattern documents determined by the matrix W_k . Relevance measure of the new document is a norm of the matrix H_{new} . In our research, we use $\|\cdot\|_\infty$, which is the maximum absolute column sum of the matrix. The higher values the elements of the matrix H_{new} have, the better the text of the considered document is characterized by the topics of the pattern document.

3 Experimental Evaluation

To estimate the performance of the suggested approach, we run several experiments on the benchmark Ansar1 dataset [8]. It was transformed to the set of documents, each of which corresponds to a thread of the forum. Small documents with the size less than 5 Kb are excluded. Ten documents from Ansar1 of the most size are taken as pattern documents. We assume that the possibility of discovering more meaningful keywords is higher in a large document. Besides, top largest documents do contain extremist information in Ansar1 dataset. The rest of Ansar1 documents are used for retrieval as "positive" examples, potentially containing extremist information. We use *talk.politics.misc*, *talk.politics.guns*, *talk.politics.mideast*, *talk.religion.misc*, *alt.atheism*, *soc.religion.christian* documents from well-known 20 Newsgroups [9] dataset as "negative" examples that definitely do not contain extremist information. We selected these themes to make experiment more real, because messages in these themes use vocabulary similar to Ansar1 dataset. As we checked the intersection is more than 50% for English vocabulary words in Ansar1 and selected Newsgroups datasets. We considered only

words that appeared at least in 10 documents. See Table 1 for the characteristics of the formed sets of documents.

Table 1. Dataset characteristics.

Dataset purpose	Dataset name	Documents
Pattern documents	Ansar1	10
“Positive” documents for retrieval	Ansar1	2732
“Negative” documents for retrieval	20 Newsgroups	5049

Experiments are carried out for each of the pattern documents according to the following scenario:

- Step 1. Extract keywords from the pattern document.
- Step 2. Choose documents with at least 10 keywords in the whole set (both “positive” + “negative” examples).
- Step 3. Rank chosen documents using topic model relevance measure. More relevant document are assumed to be more “positive”.

In the approach, we use the N-gram ($N = 3$) text lexeme representation. In all experiments we use fixed value of ONMF regularization parameter $\alpha = 100$ in (5). To evaluate the proposed approach, we also consider the traditional “baseline” approach in which the following standard methods are applied: stemming for processing text lexemes [5]; cosine similarity in the bag-of-words model for documents ranking [5]; NMF for keywords extraction [3, 4]. We denote this approach as “*Standard*”.

To estimate performance we use Average Precision (AP) measure which is calculated as the square under precision-recall curve, and Precision at k documents (P@k) that corresponds to the number of relevant results among top k ranked documents:

$$AP = \sum_{k=1}^n (P(k) \times \Delta r(k)), \quad (8)$$

where k is the rank in the sequence of the retrieved documents, n is the number of the retrieved documents, $P(k)$ is the precision at cut-off k in the list, and $\Delta r(k)$ is the change in recall from items $k-1$ to k . According to [10] for modern information retrieval, recall is no longer a meaningful metric, as many queries have thousands of relevant documents, and few users are interested in reading all of them. Extracted keywords for each of ten pattern documents are shown in Table 2 below.

Table 2. Extracted keywords.

Pattern document	Keywords
1	province, afghanistan, invader, mujahid, destroyed, invaders, soldiers, mujahideen, afghan, killed
2	shabaab, government, islamist, fight, islamic, shabab, somalia, somali, fighting, governments
3	afghanistan, battalion, explosive, assigned, soldier, improvised, soldiers, identifies, afghan, casualties
4	province, emirate, terrorist, district, terrorists, allah, islamic, destroy, mujahideen, mujahid
5	des, dan, de, مل اعل, СТАТЬ, مال سال ا, يم ال سا, daan, पोस्ता, ہب
6	allah, mujahid, prophet, peace, allan, battles, thing, battle, mujahids, wing
7	afghanistan, iraq, police, policemen, source, kill, roadside, bomb, afghan, killed
8	hasan, intention, nation, militant, soldier, military, hassan, soldiers, hood, fort
9	australia, police, bring, terrorist, somalia, somali, policy, ring, australian, terror
10	standing, king, allah, people, allāh, nation, action, deed, red, islam

Performance is estimated with the help of the abovementioned criteria calculated for two cases. *Filtered Documents* — AP is calculated only for documents found by keywords. *All Documents* — AP is calculated for all documents.

Obtained AP results for the described two cases are presented in Tables 3 and 4 for “*Our*” and “*Standard*” approaches, respectively.

Table 3. Filtered documents average precision.

Pattern document	Our approach		Standard approach	
	AP	#documents	AP	#documents
1	0.9969	1089	0.9914	688
2	0.9915	944	0.9887	450
3	0.9990	544	0.967	494
4	0.9968	1056	0.9957	766
5	0.8613	9	0.9804	70
6	0.9923	597	0.9735	608
7	0.9905	1218	0.9898	511
8	0.9827	517	0.9516	800
9	0.9774	359	0.9819	736
10	0.9483	833	0.8974	1142
Average	0.9737	717	0.9717	627

Table 4. All Documents (7781 documents) Average Precision.

Pattern document	Our approach	Standard approach
	AP	AP
1	0.7925	0.7358
2	0.7606	0.7188
3	0.7322	0.6420
4	0.7964	0.7582
5	0.6207	0.6692
6	0.7292	0.7374
7	0.7949	0.7030
8	0.7120	0.7304
9	0.6859	0.7471
10	0.7104	0.7124
Average	0.7335	0.7154

Analyzing the resulting AP values, we can state the following. Ranking with the proposed relevance measure based on ONMF topics mined from “N-gram vs sentence” matrix text representation model works pretty well, since AP values are high enough especially for the filtered documents. Proposed keywords extraction method works well too, because its usage for filtering documents significantly improve AP. Proposed method outperforms traditional approach, based on language-dependent stemming, bag-of-words model and cosine similarity measure.

4 Conclusions

Discovering extremist information resources on the Internet is very important and complex problem. Simple approaches based on expert specified keywords search do not work well. It happens because of several reasons, primary related to the multilingual nature, poor grammar and special efforts of authors of extremist texts to hide or code key concepts of their information. In attempt to solve this problem, we have developed the new approach based on usage of a document containing extremist information as a query pattern in three-stage scheme. This scheme includes automatic multilingual keywords extraction from the pattern document, search engine querying using these keywords, and search output results filtering and ranking using the relevance measure with pattern document. We develop a special representation model for text data, where a document is described as “N-gram vs sentence” matrix. We propose ONMF based LSA method to find hidden topics in such texts, and we develop algorithms for keywords extraction and relevance calculation using discovered hidden topics. The performance of the developed methods is experimentally evaluated on benchmark data. It is worth to note that proposed approach can be considered as quick-and-dirty solution that allows to find potentially extremist information on the Internet applying relatively small efforts. Sophisticated linguistic-based search systems tuned by professional anti-terrorist experts very likely outperform our method. Though, efforts spent to develop and support such language-dependent systems are much bigger than in our case.

Acknowledgment. This research is supported by RFFI Grant # 16-29-09555.

References

1. Landauer, T.K., Dumais, S.T.: A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychol. Rev.* **104**(2), 211–240 (1997)
2. Cavnar, W.B.: Using an N-gram-based document representation with a vector processing retrieval model. In: NIST Special Publication 500–225: Overview of the Third Text REtrieval Conference (TREC-3), pp. 269–278. DIANE Publishing Company (1995)
3. Kuang, D., Choo, J., Park, H.: Nonnegative matrix factorization for interactive topic modeling and document clustering. In: *Partitional Clustering Algorithms*, pp. 215–243 (2015)
4. Mirzal, A.: Converged algorithms for orthogonal nonnegative matrix factorizations. *arXiv Computing Research Repository* [arXiv:1010.5290v2](https://arxiv.org/abs/1010.5290v2), pp. 1–55 (2011)
5. Manning, C.D., et al.: *Introduction to Information Retrieval*. Cambridge University Press, Cambridge (2008)
6. Chisholm, E., Kolda, T.G.: *New Term Weighting Formulas for the Vector Space Method in Information Retrieval*. Computer Science and Mathematics Division, Oak Ridge National Laboratory (1999)
7. Xu, W., Liu, X.C.: Document clustering based on non-negative matrix factorization. In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 267–273. ACM (2003)
8. Zhang, Y., Zeng, S., Fan, L., Dang, Y., Larson, C.A., Chen, H.: Dark web forums portal: searching and analyzing jihadist forums. In: *IEEE International Conference on Intelligence and Security Informatics*, pp. 71–76 (2009)
9. The 20 Newsgroups data set. <http://people.csail.mit.edu/jrennie/20Newsgroups/>. Accessed 18 Aug 2017
10. Information retrieval. https://en.wikipedia.org/wiki/Information_retrieval. Accessed 18 Aug 2017

A Concept Driven Graph Based Approach for Estimating the Focus Time of a Document

Shashank Shrivastava^(✉), Mitesh Khapra, and Sutanu Chakraborti

Department of Computer Science and Engineering, Indian Institute of Technology,
Madras, Chennai 600036, India
{shashank,miteshk,sutanuc}@cse.iitm.ac.in

Abstract. Many text documents are temporal in nature, i.e., the contents of the document can be mapped to a specific time period. For example, a news article about the *Kargil War* can be mapped to the year 1999. Identifying this time period associated with the document can be useful for various downstream applications such as document reasoning, temporal information retrieval, etc. In this work, we propose a graph based approach for estimating the focus time of a document. The idea is to treat documents and years as nodes which are connected by intermediate Wikipedia concepts related to them. The focus year of a document can then be identified as the year which has the maximum influence over the document computed using the flow between the year node and the document node through all intermediate Wikipedia concept nodes. We evaluate our approach on two different datasets which were curated as a part of this work and show that our approach outperforms a state of the art method for estimating document focus time.

1 Introduction

The focus time of a document refers to the specific time period to which the content of the document refers to. For example, a document on the *Cuban Missile Crisis* would have the focus time as 1962 as all the events described in such a document would have occurred in that year. It should be obvious that the focus time of a document is different from the creation or modification time of the document. For example, one could write an article on the *Cuban Missile Crisis* in 2017 in which case the creation time of the document would be 2017 but the focus time would still be 1962. Identifying this focus time of the document could play an important role in several downstream applications such as temporal information retrieval document understanding, question answering, etc. For example, it is easy to identify that the question “Who was the Man of the Series in the ICC World Cup 2015” pertains to the year 2015. If the focus time of all the documents in the corpus is already known then the search space can be restricted to those documents which have a focus time of 2015. Of course, in some cases a document could have multiple time periods associated with it. For example, a blog on the *Greatest Cricket Matches of all Times* would presumably

span multiple years. However, in this work, we focus on documents which have a single focus time (or year) associated with them.

Previous work on estimating document focus time [4] uses a large collection of news articles to extract associations between words and time periods. For example, “Apollo” and “Armstrong” would be strongly associated with the time period “1969–1972” as these words frequently co-occur with the said time period. Once such word-year associations are computed, the focus time of a given document can be estimated by considering the word-year associations for all the words in the document and picking the year which has the strongest association with most words in the document. One drawback of this approach is that relying on words could often lead to a temporal drift. For example, the words “cuban”, “missile” and “crisis” may independently have strong associations with various years and thus add noise to the process of estimating document focus time. Note that *Cuban Missile Crisis* is a Wikipedia concept and when treated as such can be uniquely mapped to a single (or few) times periods thereby reducing the noise in the estimation. We propose to use Wikipedia concepts associated with the document (instead of words) in our graph based method as outlined below.

We propose a method for estimating the focus time of a document by constructing a concept based graph. The graph contains a single node representing the document at one end and one node each representing a specific year (or time period) at the other end. Each document and year node are connected to Wikipedia concepts related to them. The Wikipedia concepts related to a document are extracted by computing the ESA similarity [3] between a document and all Wikipedia concepts and retaining the top k concepts. The Wikipedia concepts related to a year are simply the top n concepts which co-occur with the year in a sentence containing the year. The edge weights are computed using ESA similarity or co-occurrence frequency. Once such a graph is constructed, we compute the influence of a year node over a document node as the sum of the flow across all paths between the year and the document. The year node which has the maximum influence is identified as the document focus time.

2 Related Work

Temporal Information Retrieval (TIR) [1, 2, 5, 6, 10] aims to meet the user information need by taking underlying temporal factors into account along with traditional notion of document relevance. Identifying document focus time plays an important role here. Some early work on identifying document focus time was reported in [4]. The authors use word statistics collected from a of news corpus by associating each word with absolute reference to past years. For example, if a document contains the word “tsunami” and if use of the word “tsunami” was more frequent over the time 2004–2005, then it can be assumed that the time period 2004–2005 is a good candidate for the document focus time. An important characteristic of this work is that it does not depend on the presence of temporal expression in the text. However, as explained before, associating each word with time may introduce noise with respect to actual time of text.

Document age estimation is another research problem relevant to our work, in which the creation time of an given document is to be predicted. In [8,9] a method for document dating based on language models has been proposed. This work uses corpus statistics for determining the date of a document. A *temporal language modeling* approach proposed in [7] has been used as the underlying framework. They divide a text corpus with respect to time segments based on word occurrences. Then correlation scores of a non-timestamped document with the language model of the documents present in each partition are calculated. The document is assigned the time stamp of the partition with which it has the maximum correlation score. Similarly, [11,12] use the term “burstiness” for searching a document in large time-stamped document collection. Lappas et al. [12] introduces a term burstiness model that uses discrepancy theory concepts to search for contiguous document sequences. They show the importance of temporal dimension of the data that facilitates in ranking and indexing the document. Similarly, Kotsakos et al. [11] propose a burstiness-aware method for dating a document. The authors utilize the lexical similarity as well as the burstiness of terms over time to predict the timestamp of a document. All these methods depend on the burstiness or the associations of a word over time. As mentioned before, our approach is fundamentally different in the sense that we aim at capturing the temporal concepts of the text and position these concepts onto a timeline.

3 Background

Our work anchors on the fundamental observation that humans tend to associate events (rather than words) to time. Hence using concept level description of text can lead to a more effective estimation of time. To get these concepts we use **Explicit Semantic Analysis (ESA)**. Gabrilovich et al. [3] introduced ESA that is used to find the Wikipedia articles which are semantically related to a given document. ESA represents a document as a vector in a high dimensional *Concept space*. The dimensions in this space represent the Semantic Concepts drawn from Wikipedia. Entries in the vector representing the text quantify the association with these Concepts. To motivate the importance of identifying the concept, consider the following example

“Mahatma Gandhi leads a 240-mile march from Ahmadabad to the sea to defy the British salt tax, thus launching a campaign of civil disobedience”

We will use this as a running example to explain our approach in the subsequent sections. The top 5 Wikipedia articles (or concepts) which have been identified by ESA for the given text are shown in Fig. 1. It can be observed that ESA effectively captures concepts related to the sentence. We are able to distinguish “*march*” as the protest march rather than the month “March” by identifying “Salt March” as one of the Wikipedia articles. The intuition behind finding the

concepts is that the focus time is the time with which the concepts present in the text are related to. In this particular example the concepts “*Mahatma Gandhi*”, “*Salt March*” and “*Civil disobedience*” relate to the year “1930” which is also the focus time of the text. We explain the process of capturing this relation in the next section.

4 Our Approach

Given a document D our aim is to find the focus time t of the document. To achieve this we propose an algorithm which contains the following steps:

Step 1: Finding Wikipedia Concepts Related to the Document D . As mentioned earlier, we believe that the concepts associated with a document play an important role in estimating the focus time of the document. To find these concepts we give the document D as input to the ESA algorithm. The output of the algorithm is a list of Wikipedia articles (concepts) \mathcal{C} related to the document. The concept $c_i \in \mathcal{C}$ has an ESA similarity score α_i with the document D .

Step 2: Identifying Candidate Focus Times for D . Once we have identified the concepts related to the given document we extract all dates appearing in the Wikipedia articles. The granularity for the date is year in our experiments. The set of all these dates forms our candidate set (i.e., we expect the focus time of the document to be one of these dates).

Step 3: Extracting *date-context* Associations. For a given date d , we find out all the sentences in the Wikipedia articles (corresponding to the concepts in \mathcal{C}) which contain this date. We associate each date d with the Wikipedia concepts which co-occur in the same sentence as the date. Specifically, we construct a context vector for each date by collecting all the concepts which co-occur with this date. Note that it is easy to extract these concepts as these are simply the Wikipedia hyperlinks appearing in the sentence. We refer to this as the *date-context*. A snapshot of the *date-context* extracted from Wikipedia page “Mahatma Gandhi” for some of the dates appearing in that article is shown in Fig. 1. We emphasize that for constructing the *date-context*, we consider only those articles which correspond to the concepts in \mathcal{C} as defined above.



Fig. 1. Explaining Concept Exploration for the example text from Sect. 3.

Step 4: Constructing a Date Concept Graph. We now construct a weighted, directed graph $G(V, E)$. The graph contains one vertex corresponding to the document D and one vertex corresponding to each of candidate dates d . Next, we have a node corresponding to each concept $c_i \in C$ relevant to the document D as identified by ESA. We have an edge between the document node and each of these concept nodes. The weight of the edge is equal to the ESA similarity α_i between the document D and the concept c_i . This is estimated as the projection of the document vector on the concepts in ESA concept space. Similarly, we have a node corresponding to each concepts appearing in the *date-context* of all the candidate dates. We have an edge between the date node and each of its *date-context* nodes. The weight of the edge can either be just 1 (indicating presence of an edge) or it can be proportional to the number of times the date occurs in the Wikipedia article corresponding to this context. Finally, we have edge between the document concept nodes and the *date-context* nodes and the weight of this edge is given by the ESA similarity between the *date-context* and the document. A snapshot of the graph is shown in Fig. 2.

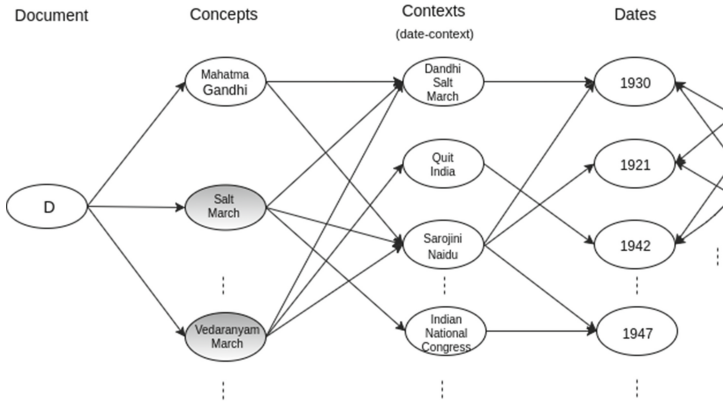


Fig. 2. Snapshot of the graph (not all nodes and edges are shown). Concept and Context are Wikipedia articles related to D , Shaded *Concept* nodes are event articles. (Contexts are also Wikipedia concepts)

Step 5: Assigning Influence Scores to Dates. Let, α_i be the weight of the edge between *concept* node c_i and the document D and β_{ij} be the weight of the edge between document *Concept* c_i and date *context* k_j . The function $f(k_j, t)$ is a function that returns the relevance score between the date *context* k_j and the *date* node t . The influence flow $I(t)$ from a date t to the document D can be computed as

$$I(t) = \sum_i^N \sum_j^M \alpha_i \times \beta_{ij} \times f(k_j, t) \tag{1}$$

where N and M are the total number of document *concepts* and time *contexts* respectively.

Step 6: Boosting Date Scores. We check if any of the concepts C related to the the document D is an event concept (i.e., it corresponds to an event page on Wikipedia). If so, we extract the date range present in the Infobox corresponding to that event page and boost the scores of all the candidate dates which lie in this range. The boosting is done by simply adding the ESA similarity scores of the event *Concept* nodes (shaded nodes in Fig. 2 are the Wikipedia event articles where the date is present in the Infobox) to the score of the date estimated by our method.

Step 7: Propagating Influence to Adjacent Dates. In the last step, we propagate the influence score computed for a given date t to its neighboring dates (years). This acts as a smoothing process, giving a score to the date nodes that gather less influence but are temporally near to a high scored *date node* (edges connecting date nodes in Fig. 2). We use a decaying function for the influence propagation so that the *date nodes* that are not adjacent will receive low activation. The propagated influence score is given by

$$I^p(t) = I(t) + \sum_{l \neq j}^L I(t_j) \times \text{prox}(t_j, t_l) \quad (2)$$

where L is the total number of years node in the graph. The function $\text{prox}(t_j, t_l)$ gives the proximity between two *Year* nodes t_i and t_l and is given by

$$\text{prox}(t_j, t_l) = 1 - \frac{|t_j - t_l|}{\text{Year}_{max} - \text{Year}_{min}} \quad (3)$$

$\text{Year}_{max} - \text{Year}_{min}$ gives the range of all the year present in the graph. Finally, the date t which has the maximum score at the end of this step is output as the document focus time.

5 Experimental Setup

In this section, we describe the datasets that were used for our experiments and the various algorithms that we compared.

5.1 Datasets

To the best of our knowledge there is no publicly available dataset for this task. In particular, the dataset used in the previous work [4] on document focus time estimation is not available publicly. To evaluate our algorithm we crawled data from two websites as described below.

1. Google Arts and Culture: This portal contains short articles which describe a particular event (typically, related to Arts and Culture). On average each article contains around 2–3 paragraphs and there is a time period (year

range) associated with the article (or event). We collected 501 such event descriptions as test data (note that our algorithm does not need any training data). These include articles ranging from year 1900 to year 1999. We found that most of the event descriptions on this website are actually taken from Wikipedia. We refer to this as the GAC dataset.

2. Historycentral.com: This website contains articles about historical events. The articles are typically shorter than the ones described earlier. Specifically, each article typically has around 2–3 sentences. Also, here each event is associated with exactly one year (and not a range of years). Further, unlike the previous dataset, the descriptions here are not taken from Wikipedia. We collected 351 such event descriptions from this website. These include articles ranging from year 1900 to year 1999. We refer to this as the HC dataset.

5.2 Models Compared

We compare the performance of the following methods:

1. EDFT: This is the model proposed in [4]. As mentioned earlier, they compute word-year associations from news articles (essentially, count the number of times a word appears in the context of a year). They then compute the document’s focus time based on the year associations of all the words in the document. They used news articles from google news archive. Unfortunately, as of this writing, articles from google news archive are no longer freely available. Further, there are restrictions in accessing them using APIs which made it difficult for us to collect these articles. To the best of our knowledge there is no other publicly available news archive which contains articles from 1900–1999 (as required by our datasets). To circumvent this issue, we used Wikipedia event pages for calculating the word-year associations as required by their algorithm. We used `petscan`¹ which allows us to find Wikipedia articles belonging to a specific category. Specifically, we collected 120 K Wikipedia event articles from 1900–1999 listed under the Wikipedia category “Events_by_decade”. We set the depth search in `petscan` to 5 which essentially means that it will search upto 5 levels of subcategories. Our search covered most of the major events and historical figures. Out of all the methods proposed by the authors the best results are obtained when the combination of *context-based association*, *temporal entropy* and *document-term frequency* is used. We report results in the result section using this combination.

2. Infobox Event Information: Here, we simply consider those ESA concepts related to the document which happen to be Wikipedia event pages. We then take the event dates mentioned in the Infobox of these articles and rank them based on the ESA similarity between the concept (corresponding to the article) and the document.

3. Graph Based Method: This is our method as described in Sect. 4. We first report the results obtained by performing only steps 1 to 5. We then report the

¹ <https://petscan.wmflabs.org>.

results obtained by adding steps 6 and 7. For rows 4,5,6 in Table 1 we set weight of edges connecting a date node i.e, $f(k_j, t)$ in Eq. 1, to 1. Finally, we report the results obtained when the weight for $f(k_j, t)$ is replaced by real valued weight proportional to the number of times the date appears in the article corresponding to the concept.

5.3 Evaluation Metrics

We use two metrics as described below for comparing the performance of different models.

1. Average Year Error: This measure was used by [4] and is computed as follows.

$$e(t_{pre}) = \begin{cases} \min\{|t_b - t_{pre}|, |t_{pre} - t_e|\} & \text{if } t_{pre} \notin [t_b, t_e] \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where t_b and t_e are the start and end years of the time period as mentioned in the ground truth data and t_{pre} is the year predicted by a given method. We report the average of the above quantity over all the test instances. Note that for the GAC data both t_b and t_e are available whereas for the HC data only a single year t_y is specified as opposed to a year range. In this case, we simply set $t_b = t_e = t_y$ and compute the error as defined above. Simply put, the year error for the HC data is just the difference between the actual year (t_y) and the predicted year.

2. Average Probability Mass Around the Ground Truth Date: If we normalize the scores assigned by a model to all the candidate years then we can treat them as a probability distribution over the candidate years. A good model should ensure that most of the probability mass is concentrated around the actual year as specified in the ground truth. We could treat the predictions of our model as a probability distribution over years. We introduce another metric which computes the percentage of the probability mass within a range of k years around the actual date. For the GAC data we compute the probability mass over all the years in the range $[t_b - k, t_e + k]$. For the HC dataset we compute the percentage of the probability mass over all the years in the range $[t_y - k, t_y + k]$.

6 Results

We now discuss the results of our experiments comparing all the models described above are summarized in Table 1.

1. Comparison with Baseline Models: Our model performs significantly better than the baseline (EDFT) model and a simple heuristic based model (Infobox event dates) which simply ranks the dates appearing in the infoboxes of event concepts related to the document based on the ESA similarity between the document and the concept. In general, the results are better for the GAC

dataset than for the HC dataset. One clear explanation for this is that most of the texts appearing in the GAC dataset are actually extracts from Wikipedia. Hence, it is easier for the ESA algorithm to find related concepts. However, this is not true for the HC dataset because of which the concepts extracted by ESA are noisy. Also, the articles in the HC dataset are shorter and hence ESA gets a smaller context as input making it harder for it to find relevant concepts. Further to speedup the time.

2. Boosting Based on Infobox Event Dates: On the GAC dataset, our basic model (steps 1 to 5) benefits from boosting based on the dates appearing in the event concepts related to the document (step 6). This is understandable because intuitively we would expect the event concepts related to a document to be more important for determining the focus time of a document. These dates extracted from the event concepts essentially reduce the noise by boosting a specific range of the dates which are important. For the HC dataset, this step actually deteriorates the performance. Once again, this is because the event concepts identified by ESA for the HC dataset are noisy and hence this step could end up boosting irrelevant candidate dates.

3. Effect of Smoothing: The smoothing done in step 7 using influence propagation on top of step 6 improves the performance by propagating the score to adjacent dates. It essentially boosts candidate dates which do not appear frequently in the text and hence do not have multiple paths to the document.

4. Using Occurrence Based Weights for Date-Context Edges: The last row of the table suggests that using co-occurrence based weights for date-context edges instead of binary weights deteriorates the performance. This is a bit counter intuitive and needs further investigations.

Table 1. Results using evaluation metric average year error (lower is better) and average probability mass (higher is better). For average probability mass, we compute the mass within a 5 year range of the true date.

Method	Average Year Error		Average Probability Mass	
	GAC	HC	GAC	HC
EDFT	10.15	19.63	44	22
Infobox Event Dates	7.8	28.36	62	19
Graph Based Method (Steps 1 to 5)	4.5	15.32	75	35
+ boosting based on Infobox (Step 6)	3.24	16.86	78	31
+ Influence propagation (Step 7)	3.03	15.81	80	38
+ real date-context weights	6.8	17.33	72	37

Table 1 presents the results based on the average mass around the ground truth date evaluation criteria. These results are obtained by incorporating

Infobox dates. This evaluates the overall quality of the estimation. To measure the statistical significance of the approaches on evaluation measure we perform paired t-test by dividing data into N number of non-overlapping folds. In both, the datasets results for all the methods are significantly better than EDFT as the p values are always less than 0.05 in all the settings. In the experiments our focus was on the effectiveness of the approach and not on efficiency, thus we have not analyzed time complexity.

7 Conclusion

In this paper, we presented a novel concept based approach for solving the problem of finding focus time of a document. We exploit the temporal relation that exists between the concepts present in the text to predict the focus time by using a graph based method. By experimenting with two new datasets, we empirically show that our method outperforms a current state of the art method thereby demonstrating that using concepts, instead of words, is more helpful in predicting the focus time. As future work, we aim to make the date-context association more robust. For example, we observed that some sentences containing dates have more than one concept associated with them. This makes it hard to associate the date with the right concept. We plan to exploit the syntactic association between the concepts and the dates present within the same sentence to make the date-context association more reliable.

References

1. Alonso, O., Strötgen, J., Baeza-Yates, R.A., Gertz, M.: Temporal information retrieval. *TWAW* **11**, 1–8 (2011)
2. Berberich, K., Bedathur, S., Alonso, O., Weikum, G.: A language modeling approach for temporal information needs. In: Gurrin, C., He, Y., Kazai, G., Kruschwitz, U., Little, S., Roelleke, T., Rüger, S., van Rijsbergen, K. (eds.) *ECIR 2010*. LNCS, vol. 5993, pp. 13–25. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12275-0_5
3. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: *IJCAI*, vol. 7, pp. 1606–1611 (2007)
4. Jatowt, A., Au Yeung, C.-M., Tanaka, K.: Estimating document focus time. In: *CIKM*, pp. 2273–2278. ACM (2013)
5. Joho, H., Jatowt, A., Roi, B.: A survey of temporal web search experience. In: *WWW*, pp. 1101–1108. ACM (2013)
6. Jones, R., Diaz, F.: Temporal profiles of queries. *ACM Trans. Inf. Syst. (TOIS)* **25**(3), 14 (2007)
7. de Jong, F.M.G., Rode, H., Hiemstra, D.: Temporal Language Models for the Disclosure of Historical Text. Royal Netherlands Academy of Arts and Sciences, Amsterdam (2005)
8. Kanhabua, N., Nørnvåg, K.: Improving temporal language models for determining time of non-timestamped documents. In: Christensen-Dalsgaard, B., Castelli, D., Ammitzbøll Jurik, B., Lippincott, J. (eds.) *ECDL 2008*. LNCS, vol. 5173, pp. 358–370. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87599-4_37

9. Kanhabua, N., Nørvåg, K.: Using temporal language models for document dating. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009. LNCS (LNAI), vol. 5782, pp. 738–741. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04174-7_53
10. Kanhabua, N., Nørvåg, K.: Determining time of queries for re-ranking search results. In: Lalmas, M., Jose, J., Rauber, A., Sebastiani, F., Frommholz, I. (eds.) ECDL 2010. LNCS, vol. 6273, pp. 261–272. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15464-5_27
11. Kotsakos, D., Lappas, T., Kotzias, D., Gunopulos, D., Kanhabua, N., and Nørvåg, K.: A burstiness-aware approach for document dating. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 1003–1006. ACM (2014)
12. Lappas, T., Arai, B., Platakis, M., Kotsakos, D., Gunopulos, D.: On burstiness-aware search for document sequences. In: Proceedings of the 15th ACM SIGKDD, pp. 477–486. ACM (2009)

Query Morphing: A Proximity-Based Approach for Data Exploration and Query Reformulation

Jay Patel and Vikram Singh^(✉)

Computer Engineering Department, National Institute of Technology, Kurukshetra 136119,
Haryana, India
jay.00174@live.com, viks@nitkkr.ac.com

Abstract. Evolution of Extremely large databases is a vital challenge for data processing via traditional database systems, such as scientific DB, Genome DB, Social Media DB etc. As these DBs are often stored in a complex schema, and inherent vastness raises challenges to a naïve user on initial data request formulation and comprehending the resulting content. A discovery-oriented search mechanism delivers good results in these information seeking scenario, as the user can stepwise explore the database and stop when the result content and quality reaches his satisfaction point. In this, understanding user's actual search intentions and how the search motives change with session progress will help greatly in achieving a search goal. A proximity-based data exploration approach, which explores the neighborhood and subsequently guides a user to overcome these limitations, named as 'Query morphing' is proposed in this paper. Various design issues and implementation constraints of the proposed approach are also listed.

Keywords: Data exploration · Hierarchical clustering
Proximity-based exploration · Query reformulation

1 Introduction

Search has become a fundamental life activity. An individual looks for meaningful information for psychological and social satisfaction. Initially, a naïve user issues short, ill-defined and imprecise queries to phrase. In turn, the search system retrieves the results based on some predefined relevance norms. This traditional *query-result* paradigm is capable to deliver sensibly, as information requests are short, and navigational and close, but not it is not always adequate [1]. Particularly, either a user is unaware of database semantics or uncertain of his exact information needs, and thus becomes challenging task for the users to phrase the informational requests [2]. Similar in discovery-oriented applications, as finding meaningful information in scientific data, genomics, health data, users requires additional help to navigate through the unknown data [3]. As user's initial search aims and intentions evolve gradually [4]. Therefore, a notion that realizes the importance of users and their intent with multiple phases of discovering, analyzing, and learning is needed; Data Exploration (DE) is one such notion. DE provides recall-oriented navigation over complex and huge datasets via short typed ill-phrased query to

precise query along with user's intent, thus requires stronger user-system interactions [5, 8].

Example-Exploratory system computes and provides results, which although not in the result set of an initial query. This allows users to explore additional information and lead towards his interest. For example, a user has searched for the movies that are directed by 'M. Scorsese' (Q_i) on the schema given in Fig. 4(a). There is a high probability that user also interested in a movie with the similar characteristic such as genre and production year, some results set are shown in Fig. 4(c). Generally, users intend carving for a wider query/data spectrum that fetches additional results for queries. Several kinds of adjustments like addition/dropping of predicate terms are required generating query variations. The various kind of variations (Q_{i+1} , shown in Fig. 4(b)) is achieved by generating morphs/variants of the initial query.

With an increase of information technology, multiple terabytes of structured and unstructured data are generated through various sources (sensors, lab simulations, social media, etc.). Due to big data occurrences, acquisition of relevant information is turned into a complex processing task. As these data are often stored in a vast and complex schema and formulating data request requires the fundamental understanding of the schema semantics. Formulating queries becomes a cognitive challenging to a naïve user. As poorly chosen or wrongly formulated queries can not only lead to huge/empty results but also get stuck in a part of the database where no satisfying results exist [6, 7, 9]. For this recurring situation, we proposed a novel proximity-based data exploration strategy that collects relevant data objects from the neighborhood of previously labeled objects. Each derived neighborhood acts as data retrieval strategy, 'Query variant/reformulation', and based on the initial query. Proposed approach primarily identifies data objects relevant to initial query in dataspace than derive neighborhood regions around each data objects. Each neighborhood-region represent region-of-interest thus treated as 'query morphs/transformation'. We named the approach '*Query Morphing*', here morphing is meant for creating a small transformation of input query. Traditionally, morphing is used to create probable transformations of input, e.g. for Image morphing [7, 35], Data Morphing [36].

There are various traditional techniques for reformulation in information retrieval (IR), as shown in Fig. 1. These techniques, the initial query submitted by information seeker goes through various transformations. The key objective of these transformations is to retrieve relevant information and improve systems performance as well [10, 11]. A user query goes through various transformations, driven by either user cognitive effort or systems assistance [12]. Query rewriting technique transforms search query in order to better represent the searchers intents, similarly, query rewriting can be viewed as a generalization of query relaxation, query expansion [12], query substitution [13] and query expansion [14]. Query substitution is modification process done based on typical possible substitutes searcher make to their query to generate new transformed query [15]. An off-the-shelf dictionary/treasure is required for all these query transformation techniques [16]. Similarly, another set of approach is in which user should be assisted for precise and unambiguous query formulation and execution. To assist users in real-time query suggestion [17] and reformulation various relevant query recommendations are generated [20]. Query suggestion can be achieved through query auto-completion

and query chain also. For recommendation and query steering [11] interactive query session is required to achieve ultimate search goal [2].



Fig. 1. Query transformations and various equivalent techniques

The traditional approach of query transformations often faces challenges of relevance and in leveraging user intent for search that recognize the importance of user participation. *Query morphing* as a proximity-based approach, is contrived as a solution for inherent challenges in data exploration of large databases. This approach mainly relies on the exploration of database and user feedback for the generation of reformulations and suggestion of the relevant objects. The data space is explored and exploited for the retrieval of the relevant data objects. We observed that the user’s query and corresponding results analogous to history log for reformulation in the process, hence it is established that query morphing will inherit the properties of traditional techniques as well.

1.1 Contribution and Outline

An algorithm for data exploration and assisted query reformation ‘*Query morphing*,’ is the key contribution of the paper. The algorithm explores the n-dimension dataspace and suggests additional relevant data objects to the user in response to the initial query. We anticipate that proposed algorithm, guides on an effective exploration over various voluminous databases. Another contribution of proposed work is an effective query reformulation approach, which is inherently supported by proximity-based data exploration.

In next section listed various related research effort and prospects. The proposed approach is discussed in Sect. 3, in which conceptual scheme depicted in a schematic diagram and algorithm. Section 4 describes various design issues and intrinsic implementation complexity in proposed approach and the analysis of implementations. Lastly, a conclusion is presented.

2 Literature Review

Next generation query processing engines should provide a much richer repertoire and easier to use querying techniques to cope with the deluge of observational data in a resource-limited setting [2, 18]. Good is good enough as an answer, provided the journey can be continued as long as the user remains interested? Searching for relevant information over huge dataset mainly affected by aspects such as Automatic exploration of data space, approximate query formulation and finally how a system assists to a user in

query formulation process. Hence, in the paper, we considered some of the prominent research works of relevant areas.

Automatic Exploration

User unable to formulate his information need in the query, which often leads to irrelevant information and a large set of returned data items. For effective results set user must be assisted throughout his exploratory session by suggesting predicates that collect all relevant information and reduce the size of returned data. Traditional DBMSs are designed for application by aiming that query is well understood by user that poses a query [1]. An application that belongs to interactive data exploration (IDE) class does not function well with this traditional DBMSs. Interactive data exploration enables a user to uncover and extract information hidden in large data through a highly ad-hoc interactive session.

Automatic Interactive Data Exploration [3] framework is developed to address such need of IDE application. AIDE integrates machine learning and data management technique to lead user towards data area of his interest. Tacking relevance feedback from user AIDE generates user exploration profile that collects sample objects and classifies data into relevant and irrelevant objects. AIDE eliminates expensive exploration quires and assists user in discovering informative and relevant data patterns. The frequency of attribute-value pair based framework like YMAL (You May Also Like) [4] can also be effective to assist user for exploration. In such techniques, computation is done based on the frequency of attribute-value pair used in query result and database instance. The assistive explorative system commonly built using relevance-feedback and models develop by facet search technique that steers user in data so that information need is fulfilled.

Query Approximation

Efficiency in data exploration is also one important concern in an explorative system when you dealing with massive amount of data. Perform several improvements in exploratory property of system without changing underlying architecture is a challenging task. Query approximation approach presents approximate result that helps in improving the response time of exploratory quires where user satisfied with 'closed-enough' answer. Several approximation techniques are developed for controlling the quality of query result where execution time is bounded and have limitations of disk and memory bandwidth. Approximation modules are designed without changing the underlying architecture like in Aqua approximate query answering system [15] that provides an approximate answer by rewriting and executing a query over summary synopsis. Statistical techniques based on synopsis [19] are widely used for Automatic Query Processing (AQP). An approximate synopsis is built to analyze large data because it's impractical to manage such big data. Four main key synopses is used is random samples, histograms, wavelet, and sketches.

A random sampling of the database space most fundament and widely used synopsis which fetched the subset of data objects based on stochastic mechanism. It's very straightforward to drawn samples from simple data table but advanced techniques are needed for big data to make sampling process scalable. BlinkDB [17] architecture a

dynamic sampling strategy that select sample based on query's accuracy and response time. Similarly, a Histogram technique summaries frequency distribution of attributes or combination of attributes and group that data values into the subset. It's also used to approximate more general class of query such as aggregation over joints. Another approach, Wavelet technique is closely worked to the histogram but the key difference is that wavelet transforms data and represent most significant into a frequency domain. AQP provide faster response time but speedup is useful when the accuracy of the returned result must be verified. Error estimation [16] and error diagnosis technique has done via bootstrap or closed forms for interactive approximate query processing ensures efficiency of the runtime as well as resource usage.

Assisted Query Formulation

Increase in growth of data availability in the day to day life enables users to peruse more and more complex information need. With the complex information retrieval, formalisms of the query are required which is mastered by a small group of adapted user. In real life user with little knowledge of querying formalisms apply brute force approach for manipulating data by hand. To resolve this issue assisted query formulation technique is used that assist user to write their queries. Several techniques are proposed to suggest terms for the incremental query formulation that minimize irrelevant data retrieval. For Boolean membership query two fundamental and critical operation equijoin and semi-join [18] are characterized to decide the tuples are formative or not in polynomial time. A learning algorithm based on user membership question [13] can also be a solution for simple Boolean query formulation. In most real-life enterprise has complex schemas and the user often unable to locate schema element of interest. Discovering query approach [25], locate minimal project join queries whose answer is similar with example tuples in output. A mastered user is often aware with the example tuples that should exist in query answer but only top-k project join query [26] requires suggesting for the better result set.

Morphing means to undergo a gradual process of transformation. Morphing is an image processing technique. In morphing process sequence of intermediate images put together with the original images that would present changes from one image to another. Similarly, in exploratory search queries should be gradually transformed to satisfy user's information need. Query morphing can be introduced as the generalized approach for several transformation techniques. A new query area formulated with the help of morphing surrounds the user request, including both past and a new variation of the query. Therefore, morphing a query gradually leads a user to a direction where information available at low cost.

3 Query Morphing: A Data Exploration Approach

Traditional lookup search is not sufficient to gain knowledge, as it retrieves best literal match optimum processing time. For this, users should aware about of '*what they are looking for*' means, must have a familiarity of the schema of database and context. In these cases search tasks are ineffective when user know what they want but may not articulate terms that would return useful results [21, 22]. Traditional search only satisfies

the information locating need but not information discovery need [23]. When user unaware or uncertain of the suitable search term for his search a traditional search technique may fails. Therefore, a system is more suitable for skill or domain expert user [24]. With the ever-increasing data volumes a serious challenge, as a naïve user has to deal with the huge set of results set in his information seeking task and complex schema. A naïve user faces challenges, primarily in the formulation of a query and intermediate queries and secondly by reviewing the query results.

Many advanced search strategies have been devised in traditional systems to narrow down the accessible information. In traditional systems, information search strongly relies on the user's ability to formulate precise queries [2, 25] and the user is expected to know how to reformulate these queries on their own. This '*Query-Result*' paradigm is inadequate for guiding a user towards relevant result via interactive exploration, as initially, a user is uncertain what exactly the information needs are [26, 29]. User's information needs evolve as the search progresses, while traditional approaches take no account of the changes in search intentions and retrieve results merely based on user queries and predefined relevance criteria. Information fetched in this way is not sufficient enough to meet user's cognitive and intellectual satisfaction. Hence, user-intent by means of feedback must be realized into fundamental search. In most of the occurrences, user feedback acts as strong relevance criteria for next search iteration [27]. This result-driven strategy is an immense help in two ways, first, it helps users to record their actual need informally into the system and second, gradually navigate through the overall search process towards more precise results. This means these systems support '*Query-Result-Review-Query*' paradigm of computing.

Three kinds of search activity lookup, learn and investigate exist. Lookup is similar to traditional search task. In lookup tasks, searcher poses a query; the system performs retrieval and return best-ranked result [37]. In this user have clear understating for what he wants in his mind and have an idea about the result. Learning is used to develop new knowledge. It aims to knowledge acquisition for understanding problem context and increases knowledge in the domain. In the investigation, the goal is to reach a decision [38]. For example, a user might perform an investigation to decide which hybrid car models to test-drive. This activity supports the investigation into a specific topic of interest. The exploratory search consists of learning activity and investigates activity [31, 37]. The goal of the traditional search is to find best in shortest time but in this we want a user to spend more time with the search for reflection. An exploratory session is of several queries where one query is a doorway for the formulation of other [32, 33]. This formulation of queries is done by morphing initial query called query point into another query point [30, 34].

3.1 Proposed Approach

The proposed approach mainly consists of two activities, first traditionally query processing and another is the generation of query variants/morphs. The exploration begins with user's submission of an initial query Q_i to query evaluation component. The validated query will be processed by a query engine and queries of subsequent interactions also processed, similarly. The major difference lies in the result visualization and

assessment. The evaluated results set consist of data objects in entire data space, these objects are relevant to search interest and thus partially fulfill user’s interest. Traditionally, an exploratory search system fetches results by considering the recall-factors of each result item and these results will be used in subsequent queries. The iterative and interactive search leads to more precise result at the end.

After processing initial query Q_i , retrieved data objects are identified in dataspace and exploited in subsequent interactions. In case of high-dimensional data space, it is assumed that, relevant results present in the close neighborhood [3, 27, 28]. Thus, exploration of a neighborhood of each object of the previous query is pivotal for futuristic search. For, each data object’s neighborhood is initialized as a cluster. A ‘Cluster-Clique’ algorithm is adapted for cluster/morph generation, Using clustering we have tried to generate possible variants with small edit distance. We want morphs of similar data which can be achieved using clustering easily. As cluster-clique is a subspace clustering algorithm we can explore morphs in any subspace means in single or overlapped subspace. It is considered that the d-dimensional spatial representation of data already created and thus partitioned into non-overlapping rectangular cells. The initial query result is projected over identified data points that identified the initial object in space, thus each data object is considered as a different unique cluster. The neighborhood of each cell containing query’s data object is explored as well as exploited to form a cluster, which covers the maximal region. A cell is dense if total data point in that cell exceeds input model parameters. Identify neighboring dense cells that form a cluster containing data point at the lower dimension (Fig. 2).

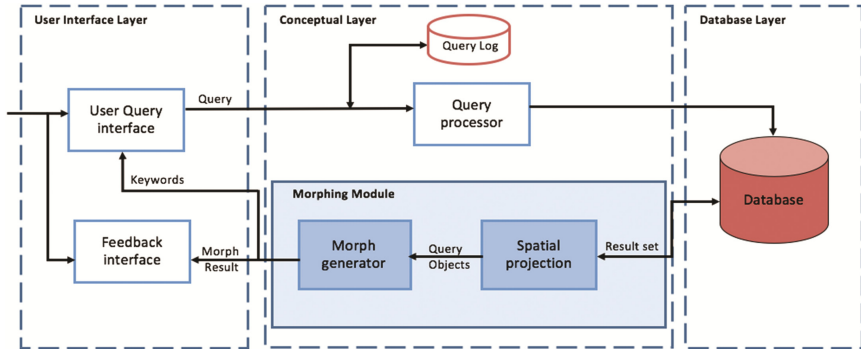


Fig. 2. Query Morphing and User’s interactions

A cluster-clique assumption is that if a k-dimensional unit is dense, then so are their projections in (k-1) dimension. Therefore, potential dense units in k-dimension can be found from (k-1) dimensional space. By examining all dense unit cluster is generated at higher dimensions. Each cluster identified is represented as morphs for our initial query. Top n keyword terms are suggested from the cluster to formulate a query for further iterative query session.

For a user posed initial query Q_i , an initial phase of the proposed system process query and returned initial result object $D \{d_{i1}, d_{i2}...d_{in}\}$. Returned result objects are

projected on d -dimension spatial representation which is partitioned into non-overlapping rectangular cells. Initially projected objects are considered as independent clusters $C = \{c_1, c_2, \dots, c_n\}$. In next step, neighborhood cells are explored and exploited if a cell is dense enough means cell contains at least t data point then merge such cells and form clusters $C = \{c_1, c_2, \dots, c_n\}$ at the lower dimension. After constructing cluster at 1-dimension next move to 2- dimension space. If there are intersecting cluster c_1 and c_2 at 1-dimension and that intersection is dense enough then merge them and form a new cluster c_{12} at 2-dimension and remove c_1 and c_2 cluster from the set. The subsequent process is done on 3rd, 4th, and up to d th dimension. Once all clusters are retrieved, we take each cluster as different morphs of the initial user query (Q_i). From all morph dataset of top- K relevant morphs with the initial data set is suggested to user for subsequent exploratory queries (Fig. 3).

Query Morph Generation Algorithm

Input: d -dimensional spatial representation of data
Output: Top- k Query morphs (M_1, M_2, \dots, M_k) and

Step 1: Identify data objects ($d_{i1}, d_{i2}, \dots, d_{in}$) for Initial user query Q_i in d -dimensional Dataspace.

Step 2: For each data point (d_{ij}),
Project data point on d -dimensional data space

Step 3: Initialize, each data point on d -dimension data space as Cluster, further Neighborhood is explored,
 {if the density of neighboring Cell is greater than *threshold* τ ,
 Then *Merge* that Cell and *Form* a Cluster at lower level }

Step 4: For each higher dimension $k \leftarrow 1$ to d
 {If, Overlapping cluster is dense at $k-1$ dimension and density is greater than *threshold* τ ,
 Then Intersection again *saved/merge* as cluster }
Merge_cluster (c_1, c_2, \dots, c_n),
 { If intersection of clusters has greater then *threshold* τ ,
 Then *Form_new_cluster*($c_{1,2,\dots,n}$),
 Remove lower dimension clusters from cluster set C }

Step 5: For each cluster C , C is equivalent to morphs M and *identify* top k -morphs,
 {*Compute* Relevance_Score of each Morphs
Identify Top- k Morphs with higher Relevance_score }

Step 6: Visualize data set of top- k morph to the user interface, End.

Fig. 3. Cluster-clique alias *Query Morphing Algorithm*

An example: Consider the movie schema, variant of the initial query ($Q_i + 1$) and the corresponding result set shown in Fig. 4. {G.genre = “Biography”} is a cluster at 1-dimension. {G.genre = “Biography”, 1990 < M.year < 2009} is cluster at 2-dimension. We are looking for interesting pieces of information at the granularity of clusters: this

may be a value of a single attribute (1-dimensional cluster) or the value of m attributes (m -dimensional cluster). Consider example query, which retrieves movies directed by ‘M. Scorsese’. User likely to interested in movies with {G.genre = “Biography”} since it is associated with many of the movies directed by ‘M. Scorsese’. The same holds for {G.genre = “Biography”, 1990 < M.year < 2009}. Besides this system also retrieve data potentially related to user need but not part of the result set of the original query. For example, consider following exploratory/variant of the initial query ($Q_i + 1$):

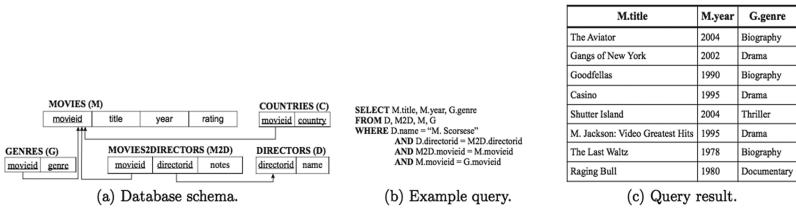


Fig. 4. (a) movie schema database (b) variant of initial query Q_{i-1} and (c) Result set of query Q_{i+1}

$(Q_{i+1}):$ **SELECT** D.name
FROM D, M2D, M, G
WHERE G.genre = “Drama” **AND** D.name <> ‘M. Scorsese’
AND D.directorid = M2D.directorid **AND** M2D.movieid=M.movieid
AND M.movieid=G.movieid.

Retrieve other movie directors that have also directed *drama* movies, which may be an interesting result for the user. In the proposed approach, these query morphs/variants generated using subspace clustering and retrieve these result set from variant queries, which might belong to user interest. A system will compute dataset of the initial query shown in Fig. 4(b) and project it on the space.

Initially, all data point is treated as the initial cluster and then neighborhood is explored to form a larger cluster. In movie database, for query Q_i clusters are created for a genre and similar for year at 1-dimension. After forming the cluster at 1-dimension next step is to steers towards higher dimensions, and at 2-dimension ‘G.genre = *Biography* and 1990 < M.year < 2009’, ‘*Drama* and year > 1963’ etc. clusters are constructed. Subsequently, move to 3rd, 4th.. dth dimension in search of the relevant cluster at the higher dimension. This completes the exploration of each data subspace around the relevant objects of initial/previous query. Now each constructed cluster is equivalent to query morphs. The data items present in each morphs are considered relevant by some measure to previous query and future probable search interest. Hence, a retrieved dataset of morphs (top-K) displayed to the user based on implicit and explicit relevance. In movie database, query morphs movies of genre ‘*Biography*’ and year > 1963 are more relevant then morph with genre ‘*Thriller*’. Therefore, morphs with ‘*Biography*’ year > 1963 considered as high relevance. The system would also suggest top-K keywords from morphs like ‘*Biography*’ based on relevance with the initial query as well result set to the user for his next

exploratory/variant query formulation. As a next step, a user may shift his interest towards another query results, inspired by result variations.

4 Design Issues and Analysis

Each query morph is a transformation of original query; in which systems are trying to develop a data retrieval strategy available in neighborhood or adjacent to original query's results. Many design issues are involved in the conceptualization of the solution, as follows:

- (i) **Neighborhood region creation:** Defining the boundary of relevant data object's neighborhood is key challenge and address in various research efforts. There are many traditional techniques exist for exploring the neighborhood region of a data objects. In our approach, defining a non-overlapping boundary based on the relevance values is pivotal.
- (ii) **Generation of Query Morphs/Variants:** Each derived neighborhood represents to a region-of-interest hence to be converted as query morph/variants. Subspace clustering is used as the core for morph generation which uses grid-based approach. If data sets represented on d-dimensional spatial cells are not dense then forming cluster becomes a challenging task. Forming cluster at higher dimension may require entertaining issue such cluster overlapping.
- (iii) **Evaluation of data object's relevance:** The data object to be retrieved based on a relevance measure with user previous query and previous result as well [39]. Relevance for data objects within each cluster is evaluated and used to define the importance of the result items. Identification of statistical information to measure the relevance of each data object is key aspects, as they influence overall systems relevance accuracy.
- (iv) **Suggestion of Top-K Morphs and their result set:** In the proposed approach, each cluster created at level is analogs to query morphs/variant. For each query morphs relevance score is evaluated and used for identification of Top-K query morphs. Relevance score will be based on the relevance score of morph dataset with previous query and dataset retrieved through that query. The identification of criteria of relevance, techniques to compute the relevance score and approach of result visualization of each morph are the key issues.
- (v) **Visualization of retrieved data objects with additional Information:** It is not feasible to visualize all the data set returned by morphs, therefore, data summarization technique should be implemented. For example, relevant keywords from the morph dataset are made available to the user in a selective manner so a user can use those keywords easily for his subsequent queries.

Traditionally several kinds of adjustments can be considered to create the query variants of a user query, such as addition/dropping of predicate terms, varying constants, widening constants into ranges, joining with auxiliary tables through foreign key relationships, etc. The kind of adjustments can be statistically driven from queries ran in the past, or exploitation of database statistics gathered so far, or even intermediate results

[40]. Since we have already spent part of our time on processing Q , the intermediate query results produced along the way can also help to achieve a cheap evaluation of Q_i . Our sketched approach aligns to proximity-based query processing, but it is generalized to be driven by the neighborhood-region in combination with re-use of previously retrieved intermediate results.

Query morphing can be realized with major adjustments to the query optimizer because it is the single place where normalized edit distances can be easily applied for generation of query alternatives. The ultimate goal would be that morphing the query pulls it in a direction where information is available at low cost. Various query formulation and suggestion techniques used currently to improve users search. But all these methods are not enough to achieve an exploratory goal. Our approach is implying proximity-based data exploration, for enlarging the returned result set, in order to enable the user to rich his goal with higher accuracy. In the ideal case, it becomes even possible to spend all time T on morphed queries.

5 Conclusion

Understanding the user's actual search intentions and how the search motives change with session progress will help greatly in achieving a precise and effective system. To provide a solution, several ideas are sketched for implementation. More control is provided to the user to understand the relationship between the queries and results. Query Morphing as a query reformulation mechanism, which primarily designs to suggest additional data objects from the neighborhood of the user's query results. There multiple variant/morphs are generated and optimally selected for reformulation.

In the solution design, we observe multiple issues including (i) Neighborhood selection, (ii) Generation of Query Morphs/Variants, (iii) Suggestion of Top-K Morphs and their results, (iv) Evaluation of data object's relevance, (v) Visualization of retrieved data objects with additional Information, and (vi) Incorporating User's feedback. Our sketched approach aligns to proximity-based query processing, but it is generalized to be driven by the query edit distance in combination with statistics and re-use of intermediate results. It could be realized with major adjustments to the query optimizer. It can also use the plan generated for Q to derive the morphed ones. The ultimate goal would be that morphing the query pulls it in a direction where information is available at low cost.

References

1. White, R.W., Roth, R.A.: Exploratory search: beyond the query-response paradigm. *Synth. Lect. Inf. Concepts Retrieval Serv.* **1**(1), 1–98 (2009)
2. Cetintemel, U., et al.: Query steering for interactive data exploration. In: 6th Biennial Conference on Innovative Data Systems Research, pp. 12–23. Asilomar, CA, USA (2013)

3. Dimitriadou, K., Papaemmanouil, O., Diao, Y.: Explore-by-example: an automatic query steering framework for interactive data exploration. In: ACM SIGMOD Conference on Management of Data, pp. 126–128. Snowbird, UT, USA (2014)
4. Drosou, M., Pitoura, E.: Ymaldb: exploring relational databases via result-driven recommendations. *VLDB J.* **22**(6), 849–874 (2013)
5. Idreos, S., Papaemmanouil, O., Chaudhuri, S.: Overview of data exploration techniques. In: ACM SIGMOD International Conference on Management of Data, pp. 277–281 (2015)
6. White, R.W., Muresan, G., Marchionini, G.: Report on ACM SIGIR 2006 workshop on evaluating exploratory search systems. *SIGIR Forum* **40**(2), 52–60 (2006). ACM
7. Kersten, M.L., et al.: The researcher’s guide to the data deluge: querying a scientific database in just a few seconds. In: *PVLDB Challenges and Visions*, vol. 3, no. 3. VLDB (2011)
8. White, R.W.: *Interactions with Search Systems*, 1st edn. Cambridge University Press, Cambridge (2016)
9. Rocchio, J.J.: Relevance feedback in information retrieval. In: Scientific Report ISR-9 (Information Retrieval) to National Science Foundation. pp. 129–140 (1971)
10. Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback. *Readings Inf. Retrieval* **24**(5), 355–363 (1997)
11. Li, H., Chan, C.Y., Maier, D.: Query from examples: an iterative, data-driven approach to query construction. In: *VLDB Endowment*, vol. 8, no. 13, pp. 2158–2169 (2015)
12. Yu, J., Qin, X.: Keyword search in databases. *Synth. Lect. Data Manag.* **1**(1), 1–155 (2009)
13. Abouzied, D., et al.: Learning and verifying quantified Boolean queries by example. In: 32nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database (PODS), pp. 49–60, ACM, New York, New York, USA (2013)
14. Abouzied, J., Hellerstein, M., Silberschatz, A.: Playful query specification with dataplay. *Very Large Data Bases Endowment (PVLDB)* **5**(12), 1938–1941 (2012)
15. Acharya, S., et al.: The aqua approximate query answering system. In: *ACM SIGMOD Record*, vol. 28, no. 2, pp. 574–576. ACM (1999)
16. Agarwal, S., et al.: Knowing when you’re wrong: building fast and reliable approximate query processing systems. In: *ACM SIGMOD Conference on Management of Data*, pp. 481–492. ACM, Snowbird, Utah, USA (2014)
17. Agarwal, S., et al.: BlinkDB: queries with bounded errors and bounded response times on very large data. In: 8th European Conference on Computer Systems (EuroSys), pp. 29–42. ACM, NY, USA (2013)
18. Bonifati, R., Staworko, S.: Interactive inference of join queries. In: 17th International Conference on Extending Database Technology (EDBT), pp. 451–462. Athènes, Greece (2014)
19. Cormode, G., et al.: Synopsis for massive data: Samples, histograms, wavelets, sketches. *Found. Trends Databases* **4**(3), 1294–1319 (2012)
20. Fan, J., Li, G.: Interactive SQL query suggestion: making databases user-friendly. In: *International Conference on Data Engineering (ICDE)*, pp. 126–136, IEEE, Hannover, Germany (2011)
21. Hellerstein, J.M., et al.: Interactive data analysis: the control project. *Computer* **32**(8), 51–59 (1999)
22. Hellerstein, J.M., et al.: Online aggregation. In: *ACM SIGMOD Record*, vol. 26, no. 2, pp. 171–182. ACM, New York, NY, USA (1997)
23. Qarabaqi, B., Riedewald, M.: User-driven refinement of imprecise queries. In: 30th International Conference on Data Engineering (ICDE), pp. 916–926. IEEE, USA (2014)
24. Sellam, T., Kersten, M.: Meet Charles, big data query advisor. In: *Biennial Conference on Innovative Data Systems Research*, pp. 94–102. Asilomar, California (2013)

25. Shen, Y., et al.: Discovering queries based on example tuples. In: SIGMOD Conference on Management of Data, pp. 493–504. ACM, Snowbird, Utah, USA (2014)
26. Psallidas, F., et al.: Top-k spreadsheet-style search for query discovery. In: SIGMOD Conference on Management of Data, pp. 2001–2016, ACM, Melbourne, Australia (2015)
27. Peng, Y.: A system for query, analysis and visualization of a multi-dimensional relational database (Doctoral dissertation) (2002)
28. Chau, D.H., et al.: Apollo: making sense of large network data by combining rich user interaction and machine learning. In: SIGCHI Conference on Human Factors in Computing Systems, pp. 167–176. ACM, Vancouver, BC, Canada (2011)
29. Ahn, J.W., Brusilovsky, P.: Adaptive visualization for exploratory information retrieval. *Inf. Process. Manag.* **49**(5), 1139–1164 (2013)
30. Ruotsalo, T., et al.: Directing exploratory search with interactive intent modeling. In: 22nd ACM International Conference on Information & Knowledge Management, pp. 1759–1764. ACM, CA, USA (2013)
31. Glowacka, D., et al.: Directing exploratory search: reinforcement learning from user interactions with keywords. In: International Conference on Intelligent User Interfaces, pp. 117–128. ACM, Tokyo, Japan (2013)
32. Ruotsalo, T., et al.: Interactive intent modeling: information discovery beyond search. *Commun. ACM* **58**(1), 86–92 (2015)
33. Klouche, K., et al.: Designing for exploratory search on touch devices. In: 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 4189–4198. ACM, Seoul (2015)
34. Andolina, S., et al.: Intentstreams: smart parallel search streams for branching exploratory search. In: 20th International Conference on Intelligent User Interfaces, pp. 300–305. ACM, Georgia, USA (2015)
35. Beier, T., Neely, S.: Feature-based image metamorphosis. In: SIGGRAPH 92, In: Computer Graphics, pp. 35–42 (1992)
36. Richard A., Jignesh, M.: Data morphing: an adaptive, cache-conscious storage technique. In: 29th International conference on very large database, pp. 417–428. Berlin (2003)
37. Dhankar, A., Singh, V.: A scalable query materialization algorithm for interactive data exploration. In: 4th IEEE International Conference on Parallel, Grid and Distributed Computing, pp. 128–133. IEEE, India (2016)
38. Singh, V., Jain, S.K.: A progressive query materialization for interactive data exploration. In: 1st International Workshop, Social Data Analytics and Management (SoDAM'2016) Co-located at 44th VLDB'2016, pp. 1–10. VLDB, India (2016)
39. Andolina, S., Klouche, K., Cabral, D., Ruotsalo, T., Jacucci, G.: InspirationWall: supporting idea generation through automatic information exploration. In: ACM SIGCHI Conference on Creativity and Cognition, pp. 103–106. ACM (2015)
40. Zhang, Y., Gao, K., Zhang, B., Li, P.: *TimeTree*: a novel way to visualize and manage exploratory search process. In: Stephanidis, C. (ed.) HCI 2016. CCIS, vol. 617, pp. 313–319. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40548-3_53

WikiSeeAlso: Suggesting Tangentially Related Concepts (*See also links*) for Wikipedia Articles

Sahiti Labhishetty^(✉), Ayesha Siddiqa, Rajivteja Nagipogu,
and Sutanu Chakraborti

Indian Institute of Technology Madras, Chennai, India
sahitilucky21@gmail.com, m.ayeshasiddiqa@gmail.com,
rajivpensidpri@gmail.com, sutanuc@cse.iitm.ac.in

Abstract. Wikipedia is the pervasive knowledge source for widely utilized applications like Google’s Knowledge Graph, IBM’s Watson and Apple’s Siri system. Wikipedia articles contain internal links and *See also* section links. According to Wikipedia, one of the purposes of *See also* links is to enable readers to explore tangentially related topics. Currently, Wikipedia relies on human judgments for adding *See also* links. We attempt to automate the process of *See also* recommendation by utilizing the aspects of Wikipedia articles like category knowledge, Backlink and the ESA concept vector similarity and external knowledge retrieved by web search engine. Our proposed ensemble based approach combines similarities obtained from these aspects to give a final prediction score. We evaluate our approach on datasets of Wikipedia articles and present our empirical comparison and case studies results with the state-of-the-art approaches. We envisage that this work will aid Wikipedia editors and readers to facilitate information search.

1 Introduction

Wikipedia is a prime example of collaboratively created and maintained content repository with English Wikipedia alone containing 5 million articles. Widespread applications like Google’s Knowledge Graph, IBM’s Watson and Apple’s Siri system rely on Wikipedia for knowledge acquisition. Consequently, Wikipedia’s content reaches millions of users and thus its content needs to be timely and accurate. Wikipedia relies on editors or readers for maintenance, improvement of its articles and for information search. However, owing to the huge size and growth rate of Wikipedia, a manual search for information is unworkable in the long run. We address this issue through automated suggestion of *See also*’s.

Wikipedia quotes that the links in the *See also*¹ section might be only indirectly related to the topic of the article because one purpose of *See also* links is to enable readers to explore tangentially related topics. Subsequently, these links can act as pointers to relevant literature. We believe that adding those links

¹ https://en.wikipedia.org/wiki/See_also.

gives an opportunity to the reader to explore more. Consequently, these links might result in expanding the breadth of the Wikipedia article.

We try to motivate the use of tangentially related topics with the following scenario. For example, consider a Wikipedia article on *Kruskal's algorithm*, which is an algorithm to solve *Minimum Spanning Tree* problem. Other algorithms which solve the same problem like *Prim's algorithm*, *Reverse-delete algorithm* and *Boruvka's algorithm* can also be of interest to the Wikipedia readers and editors. These other algorithms can act as tangentially related topics to *Kruskal's algorithm* and as pointers to the relevant literature for Wikipedia editors. In the context of this paper, our goal is to find these pointers for a given article which can aid Wikipedia editors and can be an important tool for information search. More specifically, we address the following research problem: *Given a target Wikipedia article with its preliminary information like Content, Links, and External Knowledge from Search engines, can we identify its relevant See also links or tangentially related topics?*

Identifying *See also's* for a given article is hard since it requires domain expertise on the topic to identify its tangentially related topics. However, the final decision of whether a link can be in *See also* or *not* is subjective to the editor and depends on his common sense knowledge. In addition to this, the ground-truth of *See also* links is incomplete and an evaluation in comparison to it may not reflect the effectiveness of the approach. The problem of finding related pages in Wikipedia is posed in many variants. Solutions proposed by (Adafre et al. 2007) [1], (West et al. 2009) [2], (Noraset et al. 2014) [3] and (Siddiqa et al. 2017) [4] were focused on predicting future and missing hyperlinks. However, all the above methods focused only on recommending hyperlinks whereas our prime focus is on recommending *See also* links. (Schwarzer et al. 2016)'s [5] solved a similar problem by investigating citation and proximity of citations. To the best of our knowledge, these are the only works relevant to the problem of recommending links to Wikipedia articles.

The novelty of our method lies in coming up with various similarity measures namely *BackLink* similarity, *Concept Vector* similarity, and *Web search* similarity, which are effective in identifying *See also's* and elegantly integrating this information from various sources. *BackLink* similarity ensures the diversity required in identifying desired tangentially related topics, *Concept Vector* similarity ensures that *See also's* being tangentially related topics share similar text at a conceptual level, and *Web Search* similarity identifies web links which act as supporting evidence when sufficient information is not available within the Wikipedia. Finally, we combine these similarities using different methods like *manual weights* and *classifier based weights* which gives a score for each candidate and lastly output a ranked list of suggestions.

2 Background

Explicit Semantic Analysis (ESA): In order to obtain a concept-based representation of text, we use Explicit Semantic Analysis (ESA) proposed by

(Gabrilovich and Markovitch 2007) [6]. ESA uses Wikipedia as its source of world knowledge and estimates semantic relatedness between two text fragments. ESA takes a text fragment as input and returns a list of Wikipedia concepts as output which are weighted by the relevance of the concept to the text. ESA works on the assumption that *each article in Wikipedia article corresponds to a single concept*. An inverted index containing a mapping from words to Wikipedia articles that contain them is prebuilt and stored as a preprocessing step. To obtain the ESA representation of a text fragment each word is looked up in the ESA inverted index and then the corresponding concepts containing that particular word are retrieved. These concepts are combined to form a weighted concept vector ordered by the weights corresponding to the TF-IDF value of the word.

3 WikiSeeAlso

In order to suggest *See also* links, we use some of the most common and generic properties of the *See also*'s. As discussed in Sect. 1, *See also*'s are in general tangential concepts to the target article wherein target article is nothing but the article for which we want to suggest *See also* links. Additionally, they also satisfy the property that all these links are in some way either directly or indirectly related to the target article. We capture these properties by first collecting a set of potential tangential candidates for a target article using Wikipedia's Category tree structure and then we calculate the relevance of each these articles with the target article using various measures and combine all of them together to obtain ensemble relevance measure of a candidate article with the target article. This gives a final ranked list of candidates that have both the properties of tangential and relatedness to the target articles. *WikiSeeAlso* has two main steps: (1) Candidate Generation and (2) Candidate Ranking. We give the outline of *WikiSeeAlso* in Table 1.

Table 1. Outline of the Proposed solution for See-also links

1.	Generate candidates using the sibling articles in the corresponding categories of the target article as explained in Candidate Generation step.
2.	Prune these articles using Concept Vector Similarity i.e., articles with Concept Vector Similarity above a certain threshold are only considered as candidates for later steps.
3.	Capture relevance of candidate articles using Concept Vector, Web Corpus, and Backlink similarities. Compute these similarities as described in the corresponding similarities subsections.
4.	Since our aim is to recommend <i>See also</i> sections but not content (body) hyperlinks to the target, we remove these hyperlinks from our suggestions. Produce a ranked list of <i>See also</i> suggestions as the output

Wikipedia Category Graph (WCG) Definition: Wikipedia categories² are organized in a tree like structure called the *Wikipedia Category Graph (WCG)*. This graph captures *hyponymy* or *meronymy* relations. Wikipedia category may contain subcategories which in turn recursively contain Wikipedia articles.

3.1 Candidate Generation

All the Wikipedia articles are assigned to categories under which they can be broadly classified. Target article is the which we are suggesting *See also*'s. Target articles can belong to several categories. These other articles in the category can be connected to the category in a similar way to how the target article has been connected. Therefore, these other articles can be tangential to the target article because they might be sharing a common property in terms of their connection to the category. These other articles can be called as *sibling articles* of a target.

For instance, one of the categories assigned to *Prim's algorithm* is *Spanning Tree* since it solves the problem of *Minimum Spanning Tree*. We observed that the other algorithm solving the same problem or problems related to Spanning Tree can be found in the category *Spanning Tree* like *Kruskal's algorithm*. So, all these other articles in a category can be considered as potential tangential candidates to the target article *Prim's algorithm*. We are motivated by our above observation that tangentially related articles are typically found among sibling articles. In order to get our candidates, we hypothesize that *Tangentially related articles or See also's are generally found in articles under Wikipedia categories of the target article*.

We confine ourselves by considering only the articles under the categories of the target as potential candidates for tangentially related articles. Considering the sibling articles of the target in all the categories help us capturing the breadth of the target in these different dimensions. Considering only categories present in the target article also restricts the breadth to a certain extent, therefore limiting the broadness of the tangential concepts and also keeps it specific to a target. However, it should be noted that the categories are added by humans hence the list may not be exhaustive which limits the quality and size of the candidate set.

3.2 Candidate Ranking

We have considered three aspects for candidate ranking namely Backlink, ESA Concept Vector and Web Corpus similarities.

3.2.1 Link Based Similarity in terms of Backlink (BL)

A Backlink is a type of *inlink* to the target from the potential tangential articles. Backlink of an article *A* from *B* can be perceived as an *outlink* from *B* to *A*. The basic intuition behind Backlink is that if relevance between two concepts is very high then one of the concepts is likely to appear in the content of the other

² <https://en.wikipedia.org/wiki/Category>.

article either as a follow through or as a tangentially related or as a parallel concept. Backlink is a binary valued measure i.e., if the target article appears in *hyperlinks* of a candidate article in any form then it gets a back link score of 1, otherwise 0. We noticed that Backlinks are typically present in between two parallel concepts rather generic concepts and it is supported by our analysis performed on the dataset given in Sect. 4 for the Backlink property among *See also* links of a target. From our analysis, we observed that 63% of the times if Backlink exists, *See also* links exist as well.

3.2.2 Content Based Similarity in terms of ESA Similarity (CV)

Backlink being a pure link based measure might not be effective in identifying tangentially related concepts. This issue is predominantly observed in less enriched Wikipedia articles like *Stub* and *Start* articles which contain very few links. Therefore, we utilized a content based similarity measure called ESA Concept Vector (CV) similarity which estimates the similarity between two texts at the Wikipedia concept level rather just at the word level. By our analysis on CV similarity between *See also* links and target articles for the dataset given in Sect. 4, we observed that on an average ESA CV similarity between a *See also* link and a target article is 0.158. Further, we noticed that the average ESA CV similarity among the candidates considered in our approach is 0.09. By this, we can conclude that we are able to select candidates with at least greater than 50% of the desired similarity for providing *See also* suggestions.

3.2.3 External Knowledge in terms of Web Corpus Similarity (WS)

For scenarios where within Wikipedia is insufficient for identifying relatedness between the articles for identifying *See alsos*, we leverage *Web Corpus* similarity. One such example is a query *Constellation and stars* will have many pages relevant to both of them whereas a query like *Constellation and zebra* will have pages about either one of them but not both. Web Corpus similarity utilizes external knowledge from web pages retrieved by a search engine to estimate relatedness. The intuition behind Web Corpus Similarity is that *if two articles are related then there should be at least some web pages describing about both of them together*. Each retrieved web page will give a relevance score between the target and the candidate and we further combine all these relevance scores to give a final relevance measure. This measure is analogous to a Co-citation score between two cited documents wherein the cited documents are Wikipedia articles and the citing documents are the web pages retrieved by a search engine. We calculated Web similarity between *See also* links and target article and observed that on an average Web similarity between *See also* links and target article is 0.57. Further, we noticed that the Average Web based similarity among the candidates considered in our approach is 0.50. Thus, we can conclude that we are able to select candidates with the Average Web based similarity (i.e., 0.50) with respect to the target article which is very close to the desired similarity (i.e., 0.57) with respect to the target for providing *See also* suggestions.

3.2.4 Ensemble approach for Ranking

We use an ensemble approach to elegantly combine all the above similarity measures i.e., Concept vector similarity (CV), Backlink (BL), Web Corpus based similarity (WS) obtained from various sources. These similarities are computed between a target Wikipedia article and a list of candidates obtained from Candidate Generation step. The score obtained from the ensemble measure is used to rank the candidates of the target. This has been done in two different ways: (1) Manual weights and (2) Classifier based weights.

(1) Manual combination based ensemble: Our Manual combination ensemble approach is simply the summation of all similarities i.e., CV+BL+WS. The manual ensemble is the incorporation of all the functionalities in a single measure.

(2) Decision Tree based ensemble: In the Decision Tree ensemble approach, we use Decision Tree classifier with each similarity measure acting as a feature to classify whether a candidate is a valid *See also* link or *not*. Using these features for each candidate of target, we make a feature vector which is a composition of Backlink, Concept Vector and Web based similarities. The classifier predicts the probability for a candidate being a *See also* link and is used to rank the candidates for *See also* predictions. It can be noted that typically 5–10 *See alsos* typically exists in Wikipedia articles where as our candidate list for suggesting *See alsos* comprises of 200–300 articles (i.e., sibling articles of a target) and this leads to class imbalance. To handle this issue, we experimented with two techniques: (i) Over sampling by Synthetic Minority Over-sampling Technique (SMOTE) [7] and (ii) Cost Sensitive Decision Tree Classifier.

(i) Synthetic Minority Over-sampling Technique (SMOTE) is an over sampling technique where minority class is over sampled by creating synthetic examples of the minority class. Over sampling is done by taking each minority class sample and finding k - nearest neighbors (NN) to it. Further, the synthetic examples are incorporated along the lines joining to the k - nearest neighbors from the minority class.

(ii) Cost Sensitive Decision Tree Classifier (Cost Matrix): For our problem, we realized that False Positives can be given lesser penalty than False Negatives. This is essential since we can not afford to classify a *See also* link as a *non See also* link compared to the case of a *non See also* link being classified as a *See also* link. Therefore to incorporate this prior, we gave different costs to these values in Decision Trees and incorporated it while building the Decision Tree. Subsequently, in the Cost Sensitive Decision Tree Classifier the nodes are split in terms of the cost provided instead of the accuracy of the classifier. We experimented with different values of Cost Matrix for Cost Sensitive Classification and fined the costs on our training data.

4 Empirical Evaluation

4.1 Datasets

For all our further experiments, we curated a dataset of 150 articles chosen from different domains like *Machine Learning*, *Physics*, *Computing* and *Envi-*

ronment. This dataset consists of articles at different levels of generality. It has some generic articles like *Bird*, *Oxygen*, *DNA* etc., as well as specific articles like *AdaBoost*, *Dimensionality Reduction* etc., Wikipedia articles corresponding categories and their category members were extracted with Wikimedia API³. We perform all our experiments and analysis on this dataset. By considering such diverse set of articles, we ensured that our analysis can be generalized to other articles also and are not biased by choice of one particular domain.

4.2 Performance Measures

For the qualitative evaluation of *See also* suggestions, we use the rank-based Mean Average Precision (MAP). MAP stands for the mean average of the precision scores for a given set of queries Q . Queries in our experimental setting are nothing but the titles of Wikipedia articles.

$$\text{MAP}(Q) = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{|R_q|} \sum_{j=1}^{|R_q|} \text{Precision}(R_{q,j})$$

In the above formula, $R_{q,j}$ stands for a relevant result for query q retrieved at rank j . Standard definition of Mean Average Precision (MAP) is applied to top- k suggestions with $k = 10$ and 50 . Our second performance measure is Recall. The recall is calculated for top-10 and top-50 suggestions.

$$\text{Recall} = \frac{\text{Number of See also links present in the top-}k \text{ suggestions for all the targets}}{\text{Total number of See also's present in all the target articles}}$$

4.3 Implementation Details

4.3.1 Backlink similarity

We identified Backlinks by parsing pages of target Wikipedia articles. An alternate way could be getting Backlinks from Wikipedia link graph.

4.3.2 Concept Vector Similarity

To obtain ESA Concept Vector similarity, we use the ESA implementation provided in Descartes library⁴. We used the latest June 2017 XML dump to build the ESA inverted index. The ESA technique has to prime functionalities namely *Concept Vector* retrieval and *Semantic similarity* estimation. We utilized both the functionalities for our experiments.

³ <https://www.mediawiki.org/w/api.php>.

⁴ <http://cogcomp.cs.uiuc.edu/software/descartes/descartes-0.2/doc/README.html>.

4.3.3 Web based similarity

To obtain Web Corpus based similarity, we obtained *Google* search results for this we used Google selenium driver⁵. A combined query (*target title, candidate title*) is fired as query against the search engine and the top 20 results are obtained. Further, for each web page w_i , a count score for target and candidate is calculated which represents the relevance of the web page to target or candidate accordingly. We got the count score by the count of keywords (i.e., hyperlinks) of the target or candidate in the web page. Web similarity obtained from *Google* is then the total relevance score obtained by combining $WS_{w_i}(t, c)$ of all web pages.

$$WS_{w_i}(t, c) = \frac{x * y}{x^2 + y^2 - x * y}$$

where $x = Count_score_t(w_i)$, $y = Count_score_c(w_i)$

4.4 Results

In this section, we present our qualitative results for *See also* suggestions in Tables 2 and 3. We further compare our results with the CPA, Cocit and MLT algorithms used in (Schwarzer et al., 2016)'s work. For this comparison, we consider only the articles with the *See also* section. Table 2 presents results where variants of our method are compared among themselves for 90 articles. Table 3 presents results where our method variants are compared with CPA, Cocit and MLT. We observed that only 63 out of 100 of our test queries matched with the queries for which CPA, Cocit and MLT suggestions. Thus, for a fair comparison, we report results only on these matched queries and the comparison is done only for top-10 since these methods only output 10 suggestions. Compared to CPA, Cocit and MLT, we are doing better in terms of both MAP and Recall. Our best results are highlighted in bold. We further describe our results in Sect. 5 thorough empirical comparison and case studies.

Table 2. MAP and Recall obtained by *WikiSeeAlso* for top-50 suggestions. All the Decision Tree results reported are averaged over 10-fold cross validation.

Methods	MAP	Recall
Decision Tree (oversampling + cost matrix)	0.0327	0.2201
Decision Tree (oversampling)	0.0945	0.2200
CV + BL + WS	0.1257	0.2251

⁵ [https://en.wikipedia.org/wiki/Selenium_\(software\)](https://en.wikipedia.org/wiki/Selenium_(software)).

Table 3. MAP and Recall obtained by *WikiSeeAlso* for top-10 suggestions. All the Decision Tree results reported are averaged over 10-fold cross validation.

Methods	MAP	Recall
CV + BL + WS	0.1337	0.1438
Decision Tree (oversampling + cost matrix)	0.0216	0.0579
Decision Tree (oversampling)	0.102	0.126
CPA	0.069	0.096
MLT	0.079	0.102
Cocit	0.032	0.047

5 Discussion

5.1 Existing approaches

Prior works focused on enrichment of Wikipedia stub articles by recommending hyperlinks. (Schwarzer et al., 2016) [5]’s work studies the properties of citation based algorithms namely - Co-citation (Cocit) and Co-proximity Analysis (CPA) and the Apache Lucene MoreLikeThis (MLT) function, which is a traditional text-based similarity measure. Cocit measure is based on the intuition that if two documents are cited frequently together in other documents then they are more related. In addition to, citations CPA utilizes proximity of citations for estimating relatedness between the documents. CPA works on the assumption that if two documents are cited in closer proximity in other documents then are more related than the ones cited farther way. MLT finds documents that are like in a Vector Space Model (VSM) and ranks them by TF-IDF score. (Schwarzer et al., 2016) [5] concluded from their results that MLT performed well in identifying similarly specific articles. Complementary to MLT, CPA approach is better suited for identifying a broader spectrum of related articles and popular articles. We perform a thorough empirical comparison with (Schwarzer et al., 2016)’s [5] work, these details are provided in Sect. 4 and case studies in Sect. 5.

5.2 Empirical Comparison

Hereby, we present our empirical comparison results. Any approach designed for this is desired to be recall-centric since we are building an author aiding tool, we can not afford to miss any link which is potential enough to be a desired *See also*. Thus, we included recall as one of the performance measures for comparison. Since ours is a recall centric approach, we considered top-50 suggestions in Table 2 to ensure that most of the *See alsos* are being identified by our approach. We observed that Decision Tree with oversampling and cost matrix gave close to highest recall value with lowest MAP. This trend is as expected because we biased our Decision Tree by giving higher penalty for False Negatives (FNs) compared to False Positives (FPs) as desired for our approach. We did not analyze the other case since our goal is to make the approach recall-centric. We

observed that CV+BL+WS achieved the highest MAP and recall. This can be justified by the fact that this combination approach does not have a bias towards FNs and simply combines all the functionalities which result in its overall best performance.

To compare with the other state-of-the-art approaches like CPA, Cocit, and MLT, we considered our top-10 suggestions instead of 50 since these methods provide only top 10 of them. We observed that our methods CV+BL+WS and Decision Tree with oversampling gave highest MAP and recall compared to other approaches. We further observed that Decision Tree with oversampling and cost matrix method could not recognize desired *See alsos* in top-10 suggestions but they were provided in later suggestions because it gave a higher penalty for False Negatives (FNs) compared to False Positives (FPs) consequently. Further more, we performed statistical significance test to test how significant is the difference obtained between the previous approaches and our method for the MAP and recall values. Our results have proven to be statistically significant with paired t-test with 90% confidence interval.

5.3 Case Studies

Hereby, we present nature of our results by comparing with the results of *WikiSeeAlso*, MLT, and CPA. We observed that CPA results are in general broader in nature and more diverse from the target article whereas our results are not that broad. However, our results are some what related to the target. The broadness of our results is restricted by the broadness of the categories that have been considered to obtain candidates. In addition to this, our results are also more closer and specifically relevant to the target because of the various relevance measures we are calculating corresponding to the target as desired for *See alsos*.

We explain these characteristics in detail by example. Consider the example *Atmosphere of Earth* given in Table 4, Our method *WikiSeeAlso* gave results like *water vapor*, *carbon dioxide in earth's atmosphere*, *leaching* which are related to the Atmosphere of Earth and also match with four Actual *See alsos*. CPA results are much broader in the sense it gives elements of atmosphere and earth like *water*, *oxygen* and *soil*. Similarly, for the second example Bird, our method's results are more confined to different types of birds and other closely related articles like *crane*, *list of threatened birds of the united states* whereas CPA gives articles like *mammal*, *reptile* which can be considered as sibling articles in terms of a taxonomy. Wikipedia Category structure can be much different from a taxonomy and in case of Bird, categories of Bird are more specific to Bird. Therefore, our results are much more specific in this case but still they are able to capture one of the *See alsos*.

In complementary to CPA, MLT results are very narrow and biased towards the suggestions sharing same words. For example, suggestions where a part of the title target's title is getting matched with suggestions titles. However, *WikiSeeAlso* is not predominantly effected by this bias since we use the ESA Concept similarity which is defined at the Wikipedia concept level it ensures

that there won't be any bias on the similar words. In addition to this, our method also uses Backlink which can capture tangential concepts. Therefore, our results are broader compared to MLT.

Table 4. Top 10 results for WikiSeeAlso, CPA, MLT and Actual See also's for Atmosphere of Earth and Bird

WikiSeeAlso	CPA	MLT	Actual see also's
Atmosphere of earth			
Atmosphere	Water	Jupiter	Hypermobility (travel)
Water vapor	Oxygen	Atmosphere of uranus	co ₂ in earth's atmosphere
co ₂ in earth's atmosphere	Earth	Magnetosphere of jupiter	Atmosphere
Anacoustic zone	Carbon dioxide	Jovian infrared auroral mapper	Airshed
Maximum parcel level	Fire	Extraterrestrial atmospheres	Global dimming
Leaching (agriculture)	Nitrogen	Uranus	Water vapor
Saturation vapor density	Soil	Juno (spacecraft)	Leaching (agriculture)
Radiosonde	Temperature	Comet shoemaker-levy 9	Hydrosphere
k-index (meteorology)	Gas	Exploration of jupiter	Atmospheric electricity
Heat index	Hydrosphere	Adrastea (moon)	Aviation
Bird			
List of threatened birds of US	Chordate	Ornithurae	Animal track
Bird house	Passerine	Xiaotingia	List of threatened birds of US
Superb bird-of-paradise	Animal	Modern birds	-
Western parotia	Birdlife	Archaeornithes	-
Tit (bird)	Habitat	Alvarezsauridae	-
Fecal sac	Mammal	Maniraptoriformes	-
Crane (bird)	Species	Coelurosauria	-
Crested pigeon	Endemism	Origin of birds	-
Song-bird	Reptile	Paraves	-
Bird-of-paradise	Columbidae	Enantiornithes	-

For example, consider the queries given in the Table 4. For *Atmosphere of Earth*, MLT is biased with title *Earth* giving results like *Jupiter*, *atmosphere of Uranus* etc., whereas WikiSeeAlso results are tangentially related to the concept *Atmosphere of Earth*. Similarly, in the case of *bird*, MLT results are affected by the origin and evolution of birds present in the text of the article and gives articles like *Archaeornithes*, *Maniraptoriformes* etc., whereas WikiSeeAlso results are relevant to birds and not biased by the text.

As mentioned in Candidate Generation step, we limited our candidates to the ones belonging to the category. However, other ways of identifying “See also” links spread across various categories and studies on identifying tangentially related links for a topic could be interesting future research directions.

6 Conclusions and Future Directions

In this paper, we present a novel approach to suggest *See also* links for Wikipedia articles. We provide recommendations using the semantics of Wikipedia articles like category knowledge, Backlink as well the ESA concept text similarity and external knowledge retrieved by web. We outline implications of the approach for identifying relevant literature on a given topic. We envisage that this work will aid Wikipedia editors and readers to facilitate information search.

As a part of future work, we intend to investigate how can we extend our approach to Literature Recommendation Systems which suggest relevant literature on a given topic. Such systems can be central components of the research support tools for identifying the relevant papers and thus encourages us to develop such similar tools.

References

1. Adafre, S.F., de Rijke, M.: Discovering missing links in wikipedia. In: Proceedings of the 3rd International Workshop on Link Discovery, LinkKDD 2005, Chicago, Illinois, USA, 21–25 August 2005, pp. 90–97 (2005)
2. West, R., Precup, D., Pineau, J.: Completing wikipedia’s hyperlink structure through dimensionality reduction. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, 2–6 November 2009, pp. 1097–1106 (2009)
3. Noraset, T., Bhagavatula, C., Downey, D.: Adding high-precision links to wikipedia. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, 25–29 October 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 651–656 (2014)
4. Siddiq, A., Tendulkar, A.V., Chakraborti, S.: Wikiaug: augmenting wikipedia by suggesting credible hyperlinks. CICLing: J. Res. Comput. Sci. (2017)
5. Schwarzer, M., Schubotz, M., Meuschke, N., Breitingner, C., Markl, V., Gipp, B.: Evaluating link-based recommendations for wikipedia. In: Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries, JCDL 2016, Newark, NJ, USA, 19–23 June 2016, pp. 191–200 (2016)

6. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, 6–12 January 2007, pp. 1606–1611 (2007)
7. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002)

Integrating Knowledge Encoded by Linguistic Phenomena of Indian Languages with Neural Machine Translation

Ruchit Agrawal¹(✉), Mihir Shekhar², and Dipti Misra¹

¹ Language Technologies Research Centre, IIIT Hyderabad, Hyderabad, India
ruchit.agrawal@research.iiit.ac.in

² Data Science and Analytics Centre, IIIT Hyderabad, Hyderabad, India

Abstract. Machine Translation (MT) among Indian languages is a challenging problem, owing to multiple factors including their morphological complexity and diversity, in addition to lack of sufficient parallel data for most language pairs. Neural Machine Translation (NMT) is a rapidly advancing MT paradigm and has shown promising results for many language pairs, especially in large training data scenario. We build 110 NMT systems for translation among 11 Indian languages - the first effort in the direction of NMT for Indian languages to the best of our knowledge. Also, since the condition of large parallel corpora is not met for most Indian languages, we propose a method to employ additional linguistic knowledge which is encoded by different phenomena depicted by Indian languages; like Vibhakti, Sandhi and so on. We compare the results obtained on incorporating this knowledge with the baseline systems and demonstrate significant performance improvement. We observe that although NMT models have a strong efficacy to learn language constructs, the usage of specific features further help in improving the performance. To summarize, this paper demonstrates the use of NMT techniques for Indian languages, with an emphasis on the incorporation of specific linguistic knowledge to improve translation quality.

1 Introduction

Neural Machine Translation (NMT) has shown promising results for various language pairs and is an emerging alternative to phrase-based Statistical Machine Translation (SMT). The primary appeal of NMT lies in its ability to employ algorithms which learn linguistic rules on their own from the parallel corpus, thus making it conceptually simple and eliminating the need for complex feature engineering by providing end-to-end translation. NMT generates more fluent translation as compared to phrase based SMT systems especially on lexically rich texts. Bentivogli [3] demonstrates that NMT output contains lesser lexical errors (−17%), lesser morphology errors (−19%), and significantly lesser word order errors (−50%) than its closest competitor MT paradigms for each error type. NMT systems have achieved competitive accuracy scores under large-data training conditions for language pairs such as En → Fr (English - French) and

En \rightarrow De (English - German) [16]. However, on the other hand, NMT models are unable to extract sufficient language constructs like morphology, vocabulary and word semantics in low resource scenario.

Indian languages are extremely diverse, belonging to various language families (See Table 1), employing various scripts and spanning across a multitude of dialects. The majority of Indian languages depict a high degree of agglutination and rich morphology. These factors coupled with unavailability of large parallel corpora makes translation among Indian languages especially challenging.

Table 1. ISO-639-2 codes and language families for Indian languages. **IA:** Indo-Aryan, **DR:** Dravidian, **IE:** Indo-European

Hindi	hin	IA	Gujarati	guj	IA
Urdu	urd	IA	Marathi	mar	IA
Punjabi	pun	IA	Konkani	kon	IA
Bengali	ben	IA	Tamil	tam	DR
Telugu	tel	DR	Malayalam	mal	DR
English	eng	IE			

In this paper, we explore different techniques to build effective NMT systems for Indian languages. The proposed techniques do not require any modification to the underlying neural network architecture during various training phases. The major contributions of this paper are summarized below:-

- We build 110 baseline NMT systems for translation among Indian languages and compare their performance with state-of-the-art phrase-based SMT systems trained over the same corpus. We observe a decent performance despite the low size of training corpora.
- We propose a method which employs linguistic knowledge captured by phenomena inherent in Indian languages to improve NMT performance. We observe significant improvement in performance for all language pairs over the baseline method, and comparable performance to Phrase-Based SMT, the state-of-the-art for Indian languages.

The rest of this paper is organized as follows: Sect. 3 gives the details of our NMT architecture. Section 2 briefly describes related work. Section 4 describes the datasets and resources employed in our experiments. Section 5 gives a detailed explanation of our proposed method, the experiments conducted using this approach and the results obtained. We conclude the paper and discuss future work in Sect. 6.

2 Related Work

Recent advances in the past mainly employ statistical and rule based methods for Indian language MT. Kunchukuttan [10] uses phrase-based statistical machine

translation for Indian languages using Moses [8] for phrase extraction as well as lexicalized reordering. *Sampark* [1] is a transfer based system which uses a common lexical transfer engine for Indian language translation. Most recently, [9] uses orthographic features to produce promising results in SMT for related languages. Neural Machine Translation, proposed and enhanced by [2,4,14] has been a major breakthrough in the field of Machine Translation, by providing a simple, scalable mechanism to generate end-to-end translation without extensive feature engineering. [5] proposes the use of statistical features like translation tables and language model to improve translation quality for Chinese → English. The use of linguistic features to improve NMT was shown by [13]. While they employ generic features for the language pair English → German, we propose the incorporation of linguistic knowledge which is specific to Indian languages and not encoded by traditional generic features.

3 System Architecture

The main component of our NMT model is a single neural network trained jointly to provide end-to-end translation [2,4,6,14]. Our architecture consists of a two-layered encoder-decoder framework, comprising of bidirectional Long Short Term Memory (LSTM) units, a type of Recurrent Neural Network (RNN) [12] as shown in Fig. 1, which translates a source sentence “I eat food” into a target sentence “मैं खाना खाता हूँ”. ‘EOS’ denotes the end of the sentence.

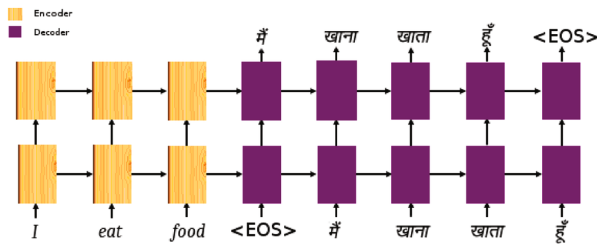


Fig. 1. A two-layered encoder-decoder based NMT architecture as proposed by [14]

The encoder encodes the source sentence into a vector from which the decoder extracts the target translation sentences. This facilitates learning of long-distance dependencies, thereby enabling the system to learn an end-to-end model. Specifically, we model the conditional probability $p(y|x)$ of translating a source sentence $x = x_1, x_2...x_u$ to a target sentence $y = y_1, y_2, ...y_v$. Let 's' be the representation of the source sentence as computed by the encoder. Based on the source representation, the decoder produces a translation, one target word at a time and decomposes the conditional probability as :

$$\log p(y|x) = \sum_{j=1}^v \log p(y_j|y_{1:j-1}, x, s) \tag{1}$$

The entire model is jointly trained to maximize the (conditional) log-likelihood of the parallel training corpus (via a softmax layer) with back-propagation through time [15].

$$\max_{\theta} \frac{1}{N} \sum_{n=1}^N \log p_{\theta}(y^{(n)}|x^{(n)}) \quad (2)$$

where $(y^{(n)}, x^{(n)})$ represents the n^{th} sentence in parallel corpus of size N and θ denotes the set of all tunable parameters.

We also use an attention mechanism [2] that allows the target decoder to look back at the source encoder.

4 Datasets and Resources

Table 1 shows the ISO-639-2 codes for various Indian languages - these are used further throughout the paper to maintain brevity. We employ the multilingual Indian Language Corpora Initiative (ILCI) corpus¹, which contains 50,000 sentences from the health and tourism domains aligned across eleven Indian languages. We employed manual preprocessing to eliminate misalignments - the resultant dataset has a size of 47,382 sentences. These are split randomly into training set, validation set and test set containing 44,000, 1382 and 2000 sentences respectively. The features used in Sect. 5.2 are generated from the intermediate outputs of *Sampark* [1] and the shallow parsers. The shallow parsers are available for nine Indian languages². We describe the features which are extracted using these in Sect. 5.2 (Table 2).

Table 2. Training, Development and Validation splits for the ILCI corpus

	Training	Validation	Test
ILCI	44000	1382	2000

5 Experiments and Results

In this section, we describe our proposed method, the related experiments and discuss the results obtained. The method employs linguistic features specific to Indian languages in addition to some generic features to improve translation quality. We use the NMT architecture as described in Sect. 3, and the tool OpenNMT [7] for its implementation. The parameters are tuned using grid search. The methodology along with the experiments are described in the following subsections. We report the results using BLEU score [11] as the evaluation metric (Table 3).

¹ This corpus is available on request from TDIL: <https://goo.gl/VHYST>.

² <https://goo.gl/Dt3zHi>.

Table 3. Corpus statistics - ILCI

	Tokens	Vocabulary
hin	850968	39170
pan	849679	849679
guj	759380	62780
tam	849679	86462
ben	715886	50553
urd	832776	36738
tel	632995	86997
kon	643605	70030
eng	808370	35134
mar	663597	77057
mal	599422	101869

5.1 Building 110 Baseline NMT Systems for Indian Languages

The ILCI parallel corpus is available for eleven Indian languages, consisting of 47,382 sentences spanning the domains of health and tourism. We train a baseline NMT model over the ILCI corpus for every possible permutation from the eleven languages, yielding 110 models. We call this model NMT_{Base} . The learning rate is set to 1 and the model is trained for 60 epochs. We employ two hidden layers and a bidirectional encoder and decoder framework.³ In order to compare our results with the state-of-the-art, we train a phrase based SMT model using the same corpus. The SMT model is trained using Moses [8] for phrase extraction and lexicalized reordering as described in [10]⁴. We call this model SMT_{SA} .

The BLEU scores obtained by NMT_{Base} are reported in Table 4. We observe slightly lower performance than Phrase-based SMT, i.e. SMT_{SA} for most language pairs. The reason behind this can be attributed to the relatively low size of the training corpus coupled with the need of NMT for relatively higher sized training datasets. We propose a method which improves performance in such low data conditions in the next section.

5.2 Exploiting Linguistic Information to Aid NMT

Large parallel corpora for Indian languages are not easily available. In order to compensate for the lack of large parallel corpora needed for NMT training, we propose the integration of linguistic knowledge encoded by specific linguistic phenomena in Indian languages as additional information to improve translation. This helps in better learning of the characteristics of Indian languages - like a

³ The detailed parameters are provided here: <http://bit.ly/2xfUj6c>.

⁴ We train our own SMT model since the training, validation and testing sets used by Sata-Anuvadak are unavailable to us.

Table 4. Baseline NMT results

	hin	urd	pan	ben	guj	mar	kok	tam	tel	mal	eng
hin	—	48.71	68.47	33.91	51.49	31.08	32.63	8.75	18.75	6.97	24.74
urd	56.88	—	51.53	24.03	38.28	16.95	22.31	5.25	11.13	4.13	16.98
pan	69.17	42.59	—	26.85	43.92	22.79	27.72	5.77	14.33	3.89	22.80
ben	35.06	22.24	29.88	—	29.26	17.70	20.32	5.04	10.46	5.69	15.79
guj	51.22	31.94	46.02	26.48	—	23.15	26.21	5.57	12.40	4.43	17.00
mar	40.94	24.02	33.14	22.28	31.70	—	25.42	5.08	8.58	3.82	12.96
kok	36.47	23.27	30.67	21.78	30.64	20.36	—	3.82	9.84	5.56	14.60
tam	19.01	11.90	15.48	11.84	13.87	7.91	10.42	—	6.69	4.01	7.62
tel	24.37	16.46	21.52	13.53	17.76	9.38	14.29	4.83	—	4.14	8.78
mal	10.51	7.02	9.23	7.06	7.17	3.48	5.04	2.14	4.27	—	5.24
eng	25.21	16.38	21.10	11.96	14.85	7.54	10.24	1.42	2.14	2.44	—

relatively free word order paradigm (with the Subject-Object-Verb (SOV) structure commonly used) where constituents of a sentence can occur in any order without affecting the overall meaning; high degree of inflection like syncretism, agglutination and allomorphy; usage of converbs; reduplication; relative participial forms and correlative clause constructions.

We extract the following features specific to Indian languages using the tools mentioned in Sect. 4. We provide examples for each feature in Hindi and Tamil, languages belonging to two different language families (Indo-Aryan and Dravidian respectively). This helps in understanding the encoding of linguistic knowledge which these features provide.

1. **Part-of-Speech tags:** Indian languages have several unique POS tags; like the following:

- Quotative
 - 'माने' ('maane', meaning 'means')
 - 'என்று' ('enru', A quotative particle)
- Demonstrative
 - 'वहाँ' ('wahaan', meaning 'there')
 - 'அந்த' ('anta', meaning 'that')
- Noun denoting spatial and temporal expressions (NST)
 - 'आगे' ('aage', meaning 'ahead' or 'further')
 - 'பின்பு' ('pinpu', meaning 'after/behind/later')
- Reduplication (RDP)
 - 'छोटे-छोटे' ('chhote (JJ) chhote (RDP)', where 'chhote' means 'small' and reduplication implies the meaning 'all of the small ones')
 - 'धीरे धीरे' ('dheere (RB) dheere (RDP)', where 'dheere' means 'slow' and the reduplication implies the meaning 'slowly, slowly')
 - 'பார்த்து பார்த்து' ('pār-tt-u (VB) pār-tt-u (RDP)', where 'pār-tt-u' means 'see' (PST-CONJ) and reduplication adds emphasis).

2. **Vibhakti** - 'Vibhakti' is a Sanskrit term for inflecting nouns and verbs, more generally used for case markers for nouns. Indian languages depict different behavior in how nouns mark their cases. Some languages have suffixes (surface case endings); some, such as Hindi, use post positions and some use a combination of the two. For example:

Vibhakti in Hindi

- (a) मनीष ने दीपा को किताब दी ।
Manish ne Deepa ko kitaab di.
manish -ne deepa -ko book give.
(Manish gave the book to Deepa.)
- (b) दीपा को मनीष ने किताब दी ।
Deepa ko Manish ne kitaab di.
Deepa -ko Manish -ne book give.
(Manish gave the book to Deepa.)

Vibhakti in Tamil

- (a) மனீஷ் தீபாவுக்கு புத்தகத்தைக் கொடுத்தான்.
Manish deepaavukku puttakattai koduttaan.
Manish deepaavu-kku puttakatt-ai koduttaan
(Manish gave the book to Deepa.)
- (b) தீபாவுக்கு புத்தகத்தைக் மனீஷ் கொடுத்தான்.
deepaavukku puttakattai maneesh koduttaan.
deepaavu-kku puttakatt-ai maneesh koduttaan
(Manish gave the book to Deepa.)

In the Hindi sentences, 'ne' and 'ko' act as the vibhakti markers. In the Tamil sentences, 'kku' and 'ai' act as the vibhakti markers. The Vibhakti information helps in mapping of semantic relations. This phenomenon allows the language to have a relatively free word-order. It can be seen that both sentences convey the same meaning, possibly with a different emphasis - which is clarified only from context. The difference in the way the markers attach to the words in Hindi and Tamil should also be noted.

3. **Sandhi** - This is a particular characteristic which leads to merging of words into a composite word. For example:

- (a) 'हिमालय' ('Himalaya', meaning 'Abode of snow') is a composite of 'हिम' ('Heem', meaning snow) and 'आलय' ('Aalay', meaning abode).
- (b) 'योगासन' ('Yogasana', meaning 'A yogic posture') is a composite of 'योग' ('Yoga', meaning union) and 'आसन' ('Asana', meaning posture).
- (c) 'சொகுசுப்பேருந்து' ('Cokucuppēruntu', meaning 'Luxury bus') is a composite of 'சொகுசு' ('cokucu', meaning comfort) and 'பேருந்து' ('pēruntu', meaning bus).

(a) and (b) are compounds and may not pose much problem. However, for highly agglutinative languages such as Dravidian languages (c); any two words can combine if the conditions for Sandhi are met, making it much more challenging to handle.

4. **Verbal inflections** - Indian languages depict significantly rich verbal inflections, which represent important grammatical information like person, number, gender, case and so on. For example, The following sentences depict the information represented by the inflections 'त्ता' ('tā'), 'गा' ('gā') and 'ள்' ('āl'):

- मैं खेलता हूँ |
main khel-tā hoon.
I play.
Here, 'tā' represents the following information: singular, male and participle.
- वह नाचेगा |
vaha naache-gā.
He will dance.
Here, 'gā' represents the information of singular, male and future tense.
- அவள் வந்தாள்
avaḷ vantaḷ.
She came.
Here 't' represents past aspect and 'āl' indicates that the subject of the action is female, singular.

Apart from the above-mentioned features, we also use two generic features - lemma and chunk heads. We train an NMT model over the ILCI corpus for 110 language pairs after performing features addition to the corpus. This is done by using piped delimitation to the OpenNMT input. We observe that feature additions helps in easier extraction of language constructs from the corpus, leading to faster learning and convergence. We call the resultant model as NMT_f . We observe a significant gain in performance over NMT_{Base} described in Sect. 5.1. Table 5 compares the results obtained by NMT_f and SMT_{SA} on the ILCI test set (We compare NMT_f with the state-of-the-art since it obtains better scores than NMT_{Base} for all language pairs).

Table 5 shows that the results obtained by NMT_f are comparable to SMT_{SA} , although NMT systems are more data hungry. We observe that although NMT models are good at learning language constructs from the parallel corpus itself, exploiting additional linguistic information in the form of features - specially in low data conditions, provides further improvement in performance.

The scores obtained by NMT_f follow a general trend - better performance for Indo-Aryan Languages and considerably poor performance for Dravidian Languages. The primary reason behind this can be attributed to larger structural similarity among Indo-Aryan Languages as well as lesser inflections as compared to Dravidian Languages, which are much more agglutinative in nature.

Table 5. Comparison of NMT_f with SMT_{SA} in terms of BLEU score

		hin	urd	pan	ben	guj	mar	kok	tam	tel	mal	eng
SMT_{SA}	hin	-	50.1	70.13	36.69	53.45	33.5	35.63	11.64	21.54	10.4	27.87
NMT_f	hin	-	51.04	71.01	36.34	53.65	33.74	35.07	10.61	20.57	8.86	27.76
SMT_{SA}	urd	57.51	-	52.3	26.36	39.08	20.7	24.84	8.36	14.9	7.92	20.64
NMT_f	urd	58.91	-	53.81	26.04	40.43	20	24.55	7.11	13.62	6.15	19.85
SMT_{SA}	pan	70.83	44.75	-	30.25	46.33	25.55	29.87	9.25	18.03	7.25	24.21
NMT_f	pan	71.59	44.87	-	29.38	46.63	24.86	30.14	7.47	16.82	5.86	24.3
SMT_{SA}	ben	36.4	24.67	31.61	-	31.13	19.84	23.18	8.68	13.6	8.94	18.44
NMT_f	ben	37.56	25.31	32.31	-	31.62	19.75	23.36	7.21	12.03	7.96	17.89
SMT_{SA}	guj	52.98	34.33	47.61	28.99	-	26.51	29.17	9.35	16.57	7.64	19.42
NMT_f	guj	53.32	34.88	48.75	28.85	-	25.83	29.3	7.86	14.59	6.04	19.62
SMT_{SA}	mar	41.97	24.99	34.51	23.89	33.54	-	27.77	8.34	12.34	7.63	16.11
NMT_f	mar	43.02	26.38	35.44	24.37	34.77	-	27.9	6.82	10.96	6.11	16.01
SMT_{SA}	kok	38.59	25.86	33.26	24.68	31.44	23.31	-	7.41	13.25	8.41	16.93
NMT_f	kok	39.15	26.01	33.53	23.87	32.76	23.4	-	5.55	11.51	7.39	17.46
SMT_{SA}	tam	21.87	15.96	19.19	14.94	17.09	11.21	14.18	-	9.13	6.61	10.7
NMT_f	tam	20.52	14.27	17.91	13.35	15.78	9.45	12.44	-	8.17	5.93	10.2
SMT_{SA}	tel	27	19.24	24.89	16.98	22.02	13.06	17.09	7.08	-	6.76	11.98
NMT_f	tel	25.98	18.19	23.65	15.36	20.24	11.81	16.13	5.97	-	5.89	10.48
SMT_{SA}	mal	13.9	10.44	12.08	10.31	10.64	7.03	8.83	4.98	6.7	-	8.2
NMT_f	mal	12.56	9.28	10.81	8.96	9.61	5.93	7.27	4.13	6	-	7.88
SMT_{SA}	eng	26.84	17.53	22.4	14.24	17.14	10.47	13.14	4.18	5.96	5.15	-
NMT_f	eng	27.24	18.94	23.19	14.76	17.83	10.56	13.31	2.95	4.34	3.76	-

6 Conclusion and Future Work

We conclude that Neural Machine Translation is an effective method to approach the challenging problem of translation among Indian languages. We demonstrate two methods to leverage NMT techniques for Indian language translation. The first method builds baseline NMT models for 110 Indian language pairs. These models obtain a decent performance on the test set. The second method comprises of employing linguistic knowledge encoded by phenomena which are specific to Indian languages to improve NMT performance. We observe significant improvement over a baseline NMT model using this method. As part of future work, we would like to work on improving the coverage of NMT models. We also observe consistently lower performance in Dravidian languages as compared to Indo-Aryan languages. We would like to specifically address the challenges faced in Dravidian language MT in the future. We would also like to explore the use of Zero Shot Neural Machine Translation for languages where no parallel corpora are available. The employment of synthetic corpora for translation among these languages is also a promising direction to work in.

References

1. Anthes, G.: Automated translation of Indian languages. *Commun. ACM* **53**(1), 24–26 (2010)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint [arXiv:1409.0473](https://arxiv.org/abs/1409.0473) (2014)
3. Bentivogli, L., Bisazza, A., Cettolo, M., Federico, M.: Neural versus phrase-based machine translation quality: a case study. arXiv preprint [arXiv:1608.04631](https://arxiv.org/abs/1608.04631) (2016)
4. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint [arXiv:1406.1078](https://arxiv.org/abs/1406.1078) (2014)
5. He, W., He, Z., Wu, H., Wang, H.: Improved neural machine translation with SMT features. In: AAAI, pp. 151–157 (2016)
6. Kalchbrenner, N., Blunsom, P.: Recurrent continuous translation models. In: EMNLP, p. 413, no. 39 (2013)
7. Klein, G., Kim, Y., Deng, Y., Senellart, J., Rush, A.M.: OpenNMT: open-source toolkit for neural machine translation. ArXiv e-prints (2017)
8. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al.: Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, pp. 177–180. Association for Computational Linguistics (2007)
9. Kunchukuttan, A., Bhattacharyya, P.: Orthographic syllable as basic unit for SMT between related languages. arXiv preprint [arXiv:1610.00634](https://arxiv.org/abs/1610.00634) (2016)
10. Kunchukuttan, A., Mishra, A., Chatterjee, R., Shah, R., Bhattacharyya, P.: Sataanuvadak: tackling multiway translation of indian languages. *pan* **841**(54,570), 4–135 (2014)
11. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics (2002)
12. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. *IEEE Trans. Sig. Process.* **45**(11), 2673–2681 (1997)
13. Sennrich, R., Haddow, B.: Linguistic input features improve neural machine translation. arXiv preprint [arXiv:1606.02892](https://arxiv.org/abs/1606.02892) (2016)
14. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
15. Werbos, P.J.: Backpropagation through time, what it does and how to do it. In: Proceedings of the IEEE, vol. 78 (1990)
16. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: Google’s neural machine translation system: bridging the gap between human and machine translation. arXiv preprint [arXiv:1609.08144](https://arxiv.org/abs/1609.08144) (2016)

Partitioned-Based Clustering Approaches for Single Document Extractive Text Summarization

Prannoy Subba¹, Susmita Ghosh^{1(✉)}, and Rahul Roy²

¹ Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032, India
prannoysubba17@gmail.com, susmitaghoshju@gmail.com

² Machine Intelligence Unit, Indian Statistical Institute, 203 B. T. Road, Kolkata 700108, India
rahulroy_r@isical.ac.in

Abstract. This article presents an extractive text summarization technique for single document using partition based clustering algorithms. Clustering of sentences is performed where the importance of each sentence in a document is attributed with three features namely, term score, keywords and average cosine similarity. Two clustering techniques, namely, k-means and fuzzy C-means are considered. To generate the summary, sentences are selected using two similarity calculation methods and the results are obtained for different compression rates (20%–60%). The results are quite promising with respect to the references used for evaluation.

Keywords: Text summarization · Sentence clustering
Extractive summarization · Similarity measures

1 Introduction

In order to tackle the ever-growing mass of available information, text summarization is in need [1–3]. Summarization of texts also stems out to be the need of the hour, as most of the information, generated today, is in the form of “streaming media” [4]. It is the process of reducing a text document and producing a condensed version while retaining the semantics of the original [1]. Single document summarization works on a single document by extracting out the key information from it [5]. The information content on it is never uniform and to find out different aspects of a single entity and extract the most important ones out of them is often a difficult task.

The very first pioneering work on rank based text summarization was carried out by Luhn [6]. In his approach, statistical values were generated by considering the word forms. The relative importance of a sentence was calculated based on the score of frequencies of words and position of sentences. Baxendale [7] introduced the sentence position feature to find out the salient parts of a document. Edmundson [8] considered four basic machine-recognizable characteristics, namely, Cue, Key, Title, and Location and assigned weights to sentences as an accumulative sum of all these four characteristic features. Other related works using statistical approaches are found in [9–12]. Linear combinations of statistical approaches as well as linguistic techniques for summarization are used in [1, 13, 14]. Recently, Fang et al. [15] introduced a word-to sentence co-rank strategy for ranking the sentences. The major issue with the ranking approach is that it

does not take into account the local relation of the sentences and provides a global view of the document.

In terms of clustering approaches towards text summarization, Zhang and Li [16] used resemblance of words between two sentences as a measure of semantic similarity between the sentences and executed k-means clustering to form similar sentence clusters. Deshpande and Lobo [17] used only the cosine similarity between the sentences to perform clustering and then used statistics based feature scores to rank each sentence in individual cluster.

To perform clustering, the use of average cosine similarity between the sentences as feature might exclude semantically related sentences which have very few words in common among them [18]. Thus to overcome this hurdle, other sentential semantics based features along with cosine similarity can be considered to better capture the semantic relation between sentences. Thereafter, clustering based approaches can be applied to form groups/clusters of semantically related sentences encompassing the entire diverseness of subjects presented in the source document. With this motivation, in the present article, an attempt is made to use two popular partitioning based clustering techniques namely, k-means and fuzzy C-means, on a feature space modeled from the sentences, for text summarization. Three features namely, term score (a function of term frequency), keywords and average cosine similarity are used to define the feature space.

To execute k-means or fuzzy C-means, the number of clusters is to be known a-priori. In our work, we have used the Elbow method [19, 20] to determine the optimal value of k (in k-means) or C (in fuzzy C-means) for carrying out sentence clustering. We have worked with a combination of three features, namely, term score, keywords and average cosine similarity to execute clustering; while most of the existing methods performed sentence clustering using only cosine similarity [17, 18]. As to the knowledge of the authors, researchers mostly used rank based approach to produce the summary; while we have employed a within cluster similarity statistic (using any one of the two complementary aspects, Jaccard similarity as well as Jaccard dissimilarity) to select the summarizing sentences.

The rest of the article is organized as follows. Section 2 describes the proposed methodology for text summarization. In Sect. 3, experimental results and their evaluation are presented. Finally, conclusion and future scope are given in Sect. 4.

2 Proposed Work

In the present work, each sentence of the source text is defined in terms of vector representation of features extracted from it. k-means and fuzzy C-means clustering algorithms are used to fragment the document into its constituent topics or subjects and then the sentences belonging to each topic are assembled. Thereafter, for each cluster, two similarity (dissimilarity) measures, namely, Jaccard similarity and Jaccard dissimilarity are used to obtain the representative sentences for the summary. Finally, based on similarity score (of sentences), a certain number of sentences are selected from each cluster (with varied compression rates,

20%–60%) to obtain the summary. The complete process is divided into several stages: pre-processing, feature extraction, learning k in k -means (or C in fuzzy C -means), clustering and post-processing. Each of them is described in more details.

2.1 Pre-processing

It involves three steps: sentence segmentation, stop word removal and word stemming. The *Break Iterator class* [21] has been used to perform sentence segmentation. Stop words (e.g., “a”, “and”, “is”, “the”, “his/her”) are the words that have little or no importance in a text, and including them would negatively influence the selection of sentences. List of stop words has been accessed from English stop word list [22]. Word stemming reduces a word into its root word form. This is done by removing all of its derivable affixes (suffixes and prefixes). In our work, the sentences are stemmed using Porter stemmer algorithm [23]. Here a sentence is segmented into its constituent words and then each word is stemmed to obtain its root form (e.g., ‘playing’ and ‘plays’ are stemmed to ‘play’).

2.2 Feature Extraction

As mentioned earlier, each sentence is modelled into a vector representing various feature values. A combination of three different features namely, term score, keywords, and average cosine similarity are used in this regard.

Term Score

Term score gives an indication of how important a term (word) is within the given text document [14, 24, 25]. To obtain it, initially the weight values of the words are computed using Eq. 1, where the frequency of a word is divided by the total number of words present in the document. Thereafter, the term score of a sentence is calculated by summing the weights of all the words present in the sentence and dividing it by the length (total number of words present) of the sentence (Eq. 2). Given a sentence S containing words w_1, w_2, \dots, w_N , the weight of a sentence S is determined by $TF(w_i)$ and $Term\ Score(S)$:

$$TF(w_i) = \frac{\text{frequency of the term}}{\text{Total number of terms in the document}} \quad (1)$$

$$Term\ Score(S) = \frac{\sum_{i=1}^N TF(w_i)}{\text{len}(S)}. \quad (2)$$

Here, N is the total number of words and $TF(w_i)$ and $\text{len}(S)$ represent the term frequency of the i^{th} word in sentence S and the length (total number of words) of S , respectively. This grades the importance of each sentence depending on how relevant the words are with respect to the document.

Keywords

Keywords are the high frequency (relevance) words present in the document. In our work, m keywords are generated from the text and a normalized frequency dependent weight value is assigned to each of it in the following manner (Eq. 3).

$$Weight(keyword_j) = \frac{TF(keyword_i)}{\sum_{i=1}^m TF(keyword_i)}, \quad (3)$$

where, m is the total number of keywords and $TF(Keyword_i)$ is the term frequency of the i^{th} Keyword. It is to be noted that the total number of generated keywords may vary with documents. In our case we have set it to 10 as it was observed during experiment that 10–15 keywords are sufficient to cover the majority of the text.

Average Cosine Similarity

Average cosine similarity determines how much similar a sentence is with respect to the rest of the sentences present in the document [24, 26]. It does so by calculating the cosine of the angle between two sentences at a time, each represented in its vector form having binary values for each word token (Eq. 4). Here, the similarity between sentences is calculated purely based on word count and not on word order.

$$\begin{aligned} \text{Avg. Cosine Similarity}(S_i) &= \frac{\sum_{j=1}^n \text{Cosine}(S_i, S_j)}{n - 1} \\ \text{Cosine}(S_i, S_j) &= \frac{S_i \cdot S_j}{|S_i| |S_j|} \end{aligned} \quad (4)$$

Here S_i and S_j represent the i^{th} and j^{th} sentence of the document and n is the total number of sentences. It is to be noted that cosine similarity provides a more precise measure of similarity and would be high only if both the sentences share a great part of their attributes. Moreover, it takes into account the variability in length of the sentences as opposed to other similarity measures (e.g., Jaccard coefficient).

2.3 Learning k in k -means

The optimum number of clusters varies with shape and scale of the distribution of sentences present in the document. In the present work, the optimal value of k in k -means (or, C in fuzzy C -means) algorithm is determined using Elbow Method [19, 20]. The Elbow Method plots the cost function for different values of k . It is noticed that the average distortion decreases with increasing value of k . However, the improvement to the average distortion declines as k increases. The value of k where the improvement to the distortion declines the most is called the elbow point (e.g., in Fig. 1, $k = 4$). It is obtained using the second derivative function on the sum of squared error (SSE) value for iterative clustering process for each value of k (Eq. 5).

$$S(x_i) = (x[i + 1] - x[i]) - (x[i] - x[i - 1])), \quad (5)$$

where $x(x_1, x_2, \dots, x_k)$ contains the SSE values for each clustering process for increasing values of k (starting with 1) and $S(x_i)$ is the second derivative for the i^{th} value of k .

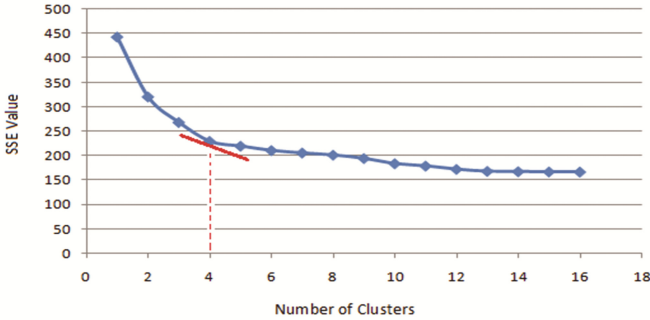


Fig. 1. Variation of SSE values with number of clusters

2.4 Clustering

In the present work, clustering is used to group similar sentences. The intuition for clustering is that it would allow capturing the topic and subtopics in a single document and provide an opportunity to exploit the locality of sentences while generating the summary. In this regard, two types of clustering algorithm viz., k-means [24, 27] and Fuzzy C-means [24, 28] are studied for capturing the topics or subject of the single document. Euclidean distance [24] measure is used for partitioning the clusters.

2.5 Post-processing

As mentioned earlier, most of the existing work used rank based approach to select the sentences for the summary. While, in our method, a cluster intrinsic similarity based approach is investigated. Post-processing involves the following steps: similarity calculation, sentence ordering, sentence selection and mapping. These are further discussed below.

Similarity Calculation

In terms of text data, the central (mean/representative) sentence of each cluster is assumed to cover a particular topic of the source document and is included in the generated summary. To select other sentences, we have explored two criteria, Jaccard similarity and Jaccard dissimilarity [29] (Eqs. 6 and 7), being complementary in nature. The reason behind considering Jaccard similarity is that the most similar sentences with respect to the central sentence are to be included in the summary. While, if Jaccard dissimilarity, a measure of contrast, is used as a criterion for sentence selection, entire diversity of a topic present within each cluster of the sample sets are assumed to be included in the summary. In our investigation, experiments are done using both in a separated manner.

$$\text{Jaccard Similarity} = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}, \quad (6)$$

$$\text{Jaccard Dissimilarity} = 1 - \text{Jaccard Similarity}, \quad (7)$$

where, S_1 and S_2 are two sentences in consideration. It may be noted that Jaccard similarity/dissimilarity measure is considered here as in case of such a measure the length of the sentences have minimal effect while selecting the sentences.

Sentence Ordering

After similarity calculation, sentence ordering is performed. Here, the sentences of each of the clusters are sorted in descending (or, ascending) order of their similarity (or, dissimilarity) values with respect to the central sentence of the cluster using Jaccard similarity (or, Jaccard dissimilarity) metric. As mentioned, the central sentence of each cluster is always included in the summary.

Sentence Selection

Based on sentence ordering, a certain specified number of sentences (depending on the compression rate) are chosen from each cluster for the summary. In our work, this ratio is taken to be in the range of 20%–60% of the original text document.

Mapping

Since each word present in sentences is in its root word form and has no semantic meaning, the selected sentences are mapped back to its original form in order to be displayed in the final summary in a human readable form. An index based mapping is performed in this regard and after mapping, the summary sentences are ordered in the same way as they appeared in original text.

The proposed clustering based text summarization method is given in Algorithm 1.

Algorithm 1.

1. *Input the text document.*
2. *Pre-process the text by extracting sentences, removing stop words and stemming of words.*
3. *Extract features from the text -- term score, keywords and average cosine similarity.*
4. *Perform k-means iteratively to determine the optimum value of k using Elbow method.*
5. *Cluster the sentences with k-means (or, fuzzy C-means) with the optimum k (or, c) value.*
6. *Order the sentences based on their similarity with respect to cluster center.*
7. *Select sentences from each cluster having similarity greater than a threshold (based on compression rate).*
8. *Output the selected sentences as summary.*

3 Evaluation and Results

To show the effectiveness of the proposed method, experiments have been conducted on different datasets and various performance measures are used for evaluation. Experiments are carried out on a machine with Intel Core i5 processor with 16 GB RAM and Windows 7 operating system. The algorithm was implemented in Java programming language.

Datasets Used

Experiment was carried out with twenty one single documents from three test files (each consisting of multiple single documents) of DUC 2001 test corpus [30]. We have also used an Essay dataset from [31] for the experiments. Results of six of the essays (of 12–200 sentences) are presented here.

Performance Metric

In this work, we have used ROUGE metric [32] to evaluate the summary generated by the proposed method. It uses precision (P), recall (R) and F-score. For each document, the method compares the candidate summary, i.e., the summary generated by our system (denoted by, $summ_{sys}$) with the reference summary (denoted by, $summ_{ref}$) and values of P, R and F-Score are computed using the following formulas. F-score parameter of ROUGE-1 (unigram co-occurrence statistics) metric is considered for evaluating the generated summary.

$$P = \frac{|Summ_{ref} \cap Summ_{sys}|}{|Summ_{sys}|}, R = \frac{|Summ_{sys} \cap Summ_{ref}|}{|Summ_{ref}|},$$

$$F - Score = 2 \frac{P * R}{P + R}.$$

As mentioned, we have worked with varied compression rates for generating summary. However, like most existing techniques, the results with 30% compression rate are reported here. For comparison purpose, the reference summary also has 30% compression rate.

The Recall, Precision, and F-Score values obtained using two partitioned based clustering methods with Jaccard similarity (denoted as, JS) and Jaccard dissimilarity (denoted as, JD) as sentence selection criteria using DUC dataset with 30% compression rate are analysed and it is noticed that for this dataset summaries obtained using Jaccard similarity metric are better as compared to those produced using Jaccard dissimilarity. Due to space limitation, the table is not presented here. However, the average results can be seen from Tables 2 and 3. k-means algorithm mostly provides better results using Jaccard similarity measure while the summary generated using both the clustering algorithms are comparable with Jaccard dissimilarity metric.

Considering F-Score values as one of the performance measuring criteria, the results obtained using the clustering algorithms with both Jaccard similarity and Jaccard dissimilarity using Essay dataset for 30% compression rate are put In Table 1. The better F-Score values with respect to sentence selection criteria are marked in bold and like the

previous dataset, it is observed that for an essay document relating to a single major subject, for majority of the cases both the clustering processes perform better when Jaccard similarity metric is used. However, here results with fuzzy C-means are found to be better than those obtained using k-means. The reason for the success of fuzzy C-means lies in the fact that it has the ability to better capture the overlap in sentences present due to semantic interrelationships in topics of the text.

Table 1. Evaluation of the summaries for Essay dataset

Essay	Clustering algorithm used	F-score	
		JS	JD
Essay 1	k-means	0.61853	0.56081
	Fuzzy C-means	0.60439	0.60453
Essay 2	k-means	0.58314	0.41478
	Fuzzy C-means	0.64079	0.41559
Essay 3	k-means	0.37846	0.50304
	Fuzzy C-means	0.59892	0.59439
Essay 4	k-means	0.67806	0.49635
	Fuzzy C-means	0.61239	0.52714
Essay 5	k-means	0.6071	0.60225
	Fuzzy C-means	0.58822	0.65707
Essay 6	k-means	0.57251	0.61742
	Fuzzy C-means	0.65874	0.57163

To analyze further, the maximum and average values (obtained over 21 single documents of DUC dataset) of recall, precision and F-score using Jaccard similarity and Jaccard dissimilarity as sentence selection criteria are depicted in Tables 2 and 3, respectively. From the tables, it is observed that for both the sentence selection strategies for all the cases k-means clustering method is providing better results as compared to its fuzzy version. However, the deviation in maximum and average values of recall, precision and F-score in k-means is more as compared to fuzzy C-means. Moreover, the average results of the two partition based clustering algorithms are comparable in nature. Thus it is evident that fuzzy C-means provides more stable clusters for text summarization as opposed to k-means. The comparable performance of fuzzy C-means in this dataset may be due to the non-overlapping nature of the topics present in it.

Table 2. Maximum and Average values (over 21 single documents) of Recall, Precision and F-Score for DUC dataset using Jaccard similarity (for sentence selection)

Algorithms used	Jaccard similarity for sentence selection					
	Recall		Precision		F-score	
	Max	Av	Max	Av	Max	Av
k-means	0.7922	0.5529	0.9143	0.5891	0.7177	0.5629
Fuzzy C-means	0.5952	0.4173	0.6831	0.4827	0.6159	0.4441

Table 3. Maximum and Average values (over 21 single documents) of Recall, Precision and F-Score for DUC dataset using Jaccard dissimilarity (for sentence selection)

Algorithms used	Jaccard dissimilarity for sentence selection					
	Recall		Precision		F-score	
	Max	Av	Max	Av	Max	Av
k-means	0.7922	0.5363	0.7143	0.5574	0.7177	0.5367
Fuzzy C-means	0.7544	0.4847	0.6327	0.4917	0.6667	0.4818

4 Conclusion

Extractive text summarization is performed with partitioning based sentence clustering methods using a combination of three features with two similarity dependent sentence extraction strategies. Our approach consists of firstly extracting out the sentence based features from the given text document. This set of features is used to perform sentence clustering using clustering algorithms (namely, k-means and fuzzy C-means). Thereafter, importance of each sentence for each topic (expressed as a cluster) in comparison to the central sentence is found out using similarity calculation methods (Jaccard Similarity and Jaccard Dissimilarity). Results are obtained for varying compression rates.

For both the datasets, Jaccard similarity mostly yields better results. Regarding the superiority of clustering algorithms, the results are contrasting in nature for the two datasets. Fuzzy C-means provide more stable clusters as compared to its crisp version.

Investigation needs to be carried out for finding out the superiority of clustering approaches using a wide variety of datasets with several other features embedded with semantic information. The scalability of the algorithm may also be investigated.

References

1. Barzilay, R., Elhadad, M.: Using lexical chains for text summarization. In: Mani, I., Maybury, M.T. (eds.) *Advances in Automatic Text Summarization*, pp. 111–121. The MIT Press, Cambridge (1999)
2. Radev, D.R., Hovy, E., McKeown, K.: Introduction to the special issue on summarization. *Comput. Linguist.* **28**(4), 399–408 (2002)
3. Manning, C.D., Schütze, H.: *Foundations of Statistical Natural Language Processing*, vol. 999. MIT press, Cambridge (1999)
4. Gupta, V., Lehal, G.S.: A survey of text summarization extractive techniques. *J. Emerg. Technol. Web Intell.* **2**(3), 258–268 (2010)
5. Andhale, N., Bewoor, L.A.: An overview of text Summarization techniques. In: *International Conference on Computing Communication Control and Automation (ICCUBEA)*, pp. 1–7. IEEE (2016)
6. Luhn, H.P.: The automatic creation of literature abstracts. *IBM J. Res. Dev.* **2**(2), 159–165 (1958)
7. Baxendale, P.B.: Machine-made index for technical literature—an experiment. *IBM J. Res. Dev.* **2**(4), 354–361 (1958)

8. Edmundson, H.P.: New methods in automatic extracting. *J. Assoc. Comput. Mach.* **16**(2), 264–285 (1969)
9. Strzalkowski, T., Stein, G.C., Wise, G.B.: A text-extraction based summarizer. In: *Proceedings of Tipster workshop*, pp. 223–230. Association for Computational Linguistics (1998)
10. Berger, A.L., Mittal, V.O.: Ocelot: a system for summarizing web pages. In: *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 144–151. ACM (2000)
11. Nomoto, T., Matsumoto, Y.: A new approach to unsupervised text summarization. In: *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 26–34. ACM (2001)
12. Radev, D.R., McKeown, K.R.: Generating natural language summaries from multiple on-line sources. *Comput. Linguist.* **24**(3), 470–500 (1998)
13. Goldstein, J., Mittal, V., Carbonell, J., Kantrowitz, M.: Multi-document summarization by sentence extraction. In: *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization*, vol. 4, pp. 40–48. Association for Computational Linguistics (2000)
14. Mani, I., Bloedorn, E.: Multi-document summarization by graph search and matching. In: *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, pp. 622–628. AAAI Press (1997)
15. Fang, C., Mu, D., Deng, Z., Wu, Z.: Word-sentence co-ranking for automatic extractive text summarization. *Exp. Syst. Appl.* **72**, 189–195 (2017)
16. Zhang, P.Y., Li, C.H.: Automatic text summarization based on sentences clustering and extraction. In: *2nd IEEE International Conference on Computer Science and Information Technology*, vol. 1, no. 1, pp. 167–170. IEEE (2009)
17. Deshpande, A.R., Lobo, L.M.R.J.: Text summarization using clustering technique. *Int. J. Eng. Trends Technol.* **4**(8), 3348–3351 (2013)
18. Skabar, A., Abdalgader, K.: Clustering sentence-level text using a novel fuzzy relational clustering algorithm. *IEEE Trans. Knowl. Data Eng.* **25**(1), 62–75 (2013)
19. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a data set via the gap statistic. *J. Roy. Stat. Soc. Ser. B (Stat. Methodol.)* **63**(2), 411–423 (2001)
20. Kodinariya, T.M., Makwana, P.R.: Review on determining number of cluster in k-means clustering. *Int. J.* **1**(6), 90–95 (2013)
21. Chan, P., Lee, R., Kramer, D.: *The Java Class Libraries: Supplement for the Java 2 Platform-Standard Edition*, v. 1.2, vol. 1. Addison-Wesley Professional, Reading (1999)
22. Bauge, K.: English stop words list. <https://sites.google.com/site/kevinbouge/stopwords-lists>
23. Porter, M.F.: An algorithm for suffix stripping. *Program* **14**(3), 130–137 (1980)
24. Bezdek, J.C.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell (1981)
25. Sarkar, K.: Sentence clustering-based summarization of multiple text documents. *Int. J. Comput. Sci. Commun. Technol.* **2**(1), 325–335 (2009)
26. Saad, S.M., Kamarudin, S.S.: Comparative analysis of similarity measures for sentence level semantic measurement of text. In: *IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pp. 90–94. IEEE (2013)
27. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297 (1967)
28. Dunn, J.C.: A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J. Cybern.* **3**(3), 32–57 (1973)

29. Niwattanakul, S., Singthongchai, J., Naenudorn, E., Wanapu, S.: Using of Jaccard coefficient for keywords similarity. In: Proceedings of the International Multiconference of Engineers and Computer Scientists, vol. 1, pp. 380–384 (2013)
30. Document Understanding Conferences (DUC). National Institute of Standards and Technology (NIST) (2001). <http://duc.nist.gov/duc2001>
31. Essay Repository. <https://www.ukessays.com/essays>
32. Lin, C.Y.: Rouge: a package for automatic evaluation of summaries. In: Proceedings of the ACL-04 Workshop on Text summarization Branches Out, pp. 74–81 (2004)

Arousal Prediction of News Articles in Social Media

Nagendra Kumar^(✉), Anusha Yadandla, K. Suryamukhi, Neha Ranabothu,
Sravani Boya, and Manish Singh

Indian Institute of Technology Hyderabad, Sangareddy 502285, India
{cs14resch11005,es14btech11024,cs17mtech01002,cs14btech11028,
es14btech11019,msingh}@iith.ac.in

Abstract. At present, news channels are using social media to disseminate news to a large audience. These news channels try to convey the news in such a way that attracts more user interaction in the form of views, likes, comments, and shares. In online news, one of the important factors in getting higher user interaction is to recommend the top news articles that would attract more number of users to give opinion, especially in the form of comments. When a news article starts getting many comments, it automatically attracts other readers to participate in the discussion. We say that a news article has “high-arousal” content if it can attract more number of comments from users. When a new news article is written it has no user interaction information, such as number of views, likes, shares or comments. In this paper, our aim is to predict news articles which have higher potential to generate high-arousal. In other words, they would attract a large number of users to give opinion in the form of comments. Unlike previous studies, we predict the arousal of news articles prior to their release, which brings the possibility of appropriate decision making to modify the article content or its ranking in audience newsfeed. We generate multiple features from the content of news articles and show that our best set of features can predict the arousal with an accuracy of 81%. We perform our experiments on social media page of CNN news channel, containing four years of data with 33,324 news articles and 226.83 million reactions.

1 Introduction

Online news channels have become one of the most popular means of mass communication. Their popularity has increased rapidly due to the extensive usage of smartphones and social media [11, 25]. News channels generate a large volume of information every day. Without a good recommendation system [1], especially personalized recommendation system [6, 13, 14], it would be too overwhelming for users to browse through all the news articles.

In this paper, we particularly take up the task of predicting arousal for online news articles published in social media. We say that a news article has “high-arousal” content if it can attract more number of users to express their opinion

on the news topic. Arousal is similar to popularity with the difference being that, arousal ensures lots of user feedback or comments. Posts with high arousal are an asset to news channels and can give a big thrust to their businesses. A news channel can become more popular if it attracts a large number of user opinion. When a news article starts getting many user opinions, it attracts other users to read the article and give their opinion. We can enhance the existing news recommendation systems if we recommend to users, articles that are relevant to their interest and also have high-arousal content.

In the traditional print media, a user does not have the option to express opinion or read opinion of other users. However, online news channels allow users to express opinion in the form of likes, shares, and comments. Likes and shares are positive reactions, where the user most often agrees with the news article. Opinion in the form of comments can be both positive and negative. Generally, comments are more aggressive form of opinion compared to likes or shares. In online news channels, users are not only interested in reading the news article, they are even more interested to read the comments of other users. In this paper, we define arousal as a function of these three user interactions, namely likes, shares, and comments. Given a new news article based on its content we predict whether the article would generate high arousal or not.

Apart from news article recommendation, there are many other advantages of showing high-arousal articles to users. This is the easiest way to get valuable user opinion, which can be mined to understand opinion of users. As e-commerce reviews are very useful in online purchase, the opinion of readers on news articles can help us to understand the opinion of population on various important issues. Often these comments give a more accurate picture of the reality compared to the original news article, which may be written in a biased manner due to various reasons. If a news article is genuine, a majority of the comments will support the article, otherwise they would disagree with the article. By improving news article recommendation, we can get more reader participation. In this paper, we address the general problem of finding whether a news article is of high-arousal or not. This can be combined with recommendations systems to recommend articles that have high-arousal and is of high interest to the reader.

Our key contributions are as follows:

- We predict arousal of news articles using social media interactions.
- We define arousal of a news article in terms of social interactions such as likes, comments, and shares.
- We perform an extensive analysis to find the suitable methods for arousal prediction.
- We conduct several experiments to find the features relevant for arousal prediction.
- We perform experiments on one of the most popular social media news channels CNN where the best approach predicts the posts of high arousal with 81% accuracy.
- Finally, we show the topics that can lead to high arousal for the news articles.

2 Related Work

Online news analysis is a popular research topic in the field of computer science, communication, and psychology [5, 8, 23]. Arousal of news articles is a subjective measure, which is defined in terms popularity metrics such as views, clicks, likes, comments, shares etc. Several related works have been proposed to forecast the popularity of news articles [2, 23, 24]. A growing number of studies [3, 24, 29] have been proposed to predict the early popularity of news articles, which has been found to have a strong correlation with the popularity of the content. All these works can be mainly categorized into two types: social connection based and content based.

Social connection based methods [20, 22, 27, 30] use social features such as number of friends, followers, etc., to predict content popularity. Zaman et al. [30] studied reaction behaviour of retweetability among users. They used author information such as number of followers, identity of the source, etc. Suh et al. [22] analyzed the factors that impact retweeting and showed that the number of followers and friends have a lot of impact, while, e.g., number of statuses and favorites do not. Petrovic et al. [20] used passive-aggressive algorithm to predict if a tweet will be retweeted which would lead to high information spread through a large number of users. Weng et al. [27] predicted future popularity of an article using its early spreading patterns. They concluded that features based on community structure are the most powerful popularity predictors.

There are several content based methods to predict the popularity of news articles [12, 19, 23, 28]. Lee et al. [12] proposed a framework that can predict the number of comments by observing an article for 2-3 days. Tatar et al. [23] proposed a method to rank the news articles by predicting users comments. Naveed et al. [19] showed that a news would be passed to other users in the network based on its content i.e., negative news spread faster than positive news. Wu et al. [28] showed that the most negative news fade rapidly and positive news fade slowly. Their study mainly focused on polarity of news article which showed that the most negative information quickly disappears and the most positive tends to persist. In contrast with existing works, we perform our analysis on social media pages of news channels where the goal is to predict the arousal of news articles before it is published in social media. We derive the features from news articles that are related to news coverage and popularity and select the features related to arousal prediction using word embeddings.

3 Methodology

We perform the following steps to determine whether a new article has high arousal or not: (1) Use unsupervised approach to create a training dataset of posts with high and low arousal; (2) Generate the candidate features; (3) Select features relevance to arousal prediction; (4) Predict posts of high arousal; (5) From posts with high arousal find the topics of high arousal. In the rest of the paper, we use terms such as ‘post’, ‘news article’, ‘news post’, and ‘news content’ interchangeably.

3.1 Finding the Posts of High Arousal

In this paper, we use supervised learning to predict post arousal. However, there is no existing dataset that has labeled posts with high and low arousal. Moreover, this kind of manually labeled training dataset is not so useful because the topics that arouse people today will not be the arousing topics after sometime. It is also difficult to label thousands of news articles manually. Thus our first step is to use an unsupervised approach to find posts that have high arousal.

Arousal of news article is a highly subjective term. We define it in terms of popularity. There are many popularity measures such as number of likes, comments, shares, clicks, views, etc. Among all these popularity measure likes, comments and shares are publicly available measures. If a post has high popularity, it means that post content is interesting enough that many people are interested in it. When many users are interested in something, it might also arouse other users to look at the content. Therefore, we use popularity as one of deciding factor in the computation of arousal. We measure the popularity score (pt_{score}) as follows:

$$pt_{score} = l + \alpha * c + \beta * s \quad (1)$$

where l , c , s are the number of likes, comments, and shares respectively. α and β are the comment and share popularity constants. As suggested by Bucher et al. [4], ‘comment’ and ‘share’ require higher cognitive effort or commitment than ‘like’. Therefore, ‘comment’ and ‘share’ outweigh ‘like’ suggesting that α , β values are greater than one. Further, ‘share’ generates higher amount of engagement compared to ‘comment’ as shared post appears on user’s profile page. A shared post is pushed towards user’s connections as it constitutes a part of user’s self-presentation. This indicates that ‘share’ outweigh ‘comment’ (or $\beta > \alpha$). In our experiment, we set the value of α , β to 2, 4 respectively which are derived from the analysis conducted by Kim et al. [10].

Having high popularity does not ensure high arousal as there are many popular posts with a large number of likes and shares, but very few comments. Likes and shares signify that users like and agree with the post content. It is through comments that the users express their agreement, disagreement or opinion with the news article. Therefore, out of the total number of reactions if comments constitute the major fraction, then we acknowledge the post as a comment dominant post. We compute the comment dominant score (cd_{score}) as follows:

$$cd_{score} = c / (l + \beta * s) \quad (2)$$

A large value of cd_{score} indicates that post has received a considerable amount of user comments. From our empirical evaluation, we found that if a post is popular and it has value of cd_{score} greater than 0.16, then it has a large number of comments. We found this number using user study. We created different labeled datasets based on different values of cd_{score} . We asked 5 users to go through the different labeled dataset and find the parameter that gives the best training sample.

Further, if we use only Eq. 2, we may get posts that have relatively a large number of comments, but they may not be much popular. In other words, posts

containing a large proportion of the comments are obtained. However, their popularity score, as defined in Eq. 1 is low. Such articles will be of interest to only a small audience. In Eq. 3, we define arousal score (a_{score}), where we exalt the comment dominant score by multiplying it with the log of popularity score.

$$a_{score} = cd_{score} * \log(pt_{score}) \quad (3)$$

a_{score} ensures that post should have a large number of comments as well as it should be popular. Logarithmic value of popularity ensures that if a post receives too many reactions, arousal is not increased too much. We rank the posts based on arousal, with the first post being the post of highest arousal and the last one being the post of lowest arousal.

3.2 Generate the Candidate Features

We use the following sets of features to classify high and low arousal news articles:

POS Tag Based Features. We use part-of-speech (POS) tagging to select features that are of interest to readers and news channels. Bandari et al. [3] showed that mentioning well-known entities in the post can increase its likelihood of becoming a popular post. For instance, consider the following news posted by CNN: “I don’t want to do that at all. Trump said: I just want what’s right.” This post received lots of user attention because it mentions a well-known entity, Mr. Trump. Interestingly, important entities like ‘Trump’ are noun features. We therefore perform POS tagging, and then select the nouns and noun phrases as entities [9]. The POS Tagger assigns part-of-speech tag to each word of the given text, such as noun, verb, adjective, etc. Nouns are represented by ‘NN’, adjectives are represented by ‘JJ’, etc. We use the Stanford POS tagger [17] to do the tagging, and then extract the words with ‘NN’, ‘NNS’, ‘NNP’, and ‘NNPS’ tags as features.

Frequency Based Features. Important issues or entities appear frequently in news articles. We use term frequency and inverse document frequency (TF-IDF) to find unigram and bigram terms that are important news topics. TF-IDF score, which is often used in text mining, shows how important a word is to a document in a corpus. The importance increases proportional to the number of times the term appears in the document (TF), but it is offset by the frequency of the word in the corpus (IDF). In our problem, we found that just using the TF score gives better result compared to using the product of TF and IDF score. The IDF score scales some terms inappropriately. Although IDF is very useful for ranking documents in information retrieval, it is not so useful in our problem. IDF gives more importance to terms that are rare in the corpus. However, to get high audience engagement we need terms that are popular among audience. Although we don’t use IDF score explicitly, we are able to remove less important words with high term frequency by using POS tags and seeing the term’s relevance to the article domain through the use of word-embedding, which is described below in Sect. 3.3.

3.3 Feature Selection

We take a combination of all the features from the previous section, and build a bag-of-words feature set, which is a high-dimensional dataset. All features obtained in the previous step may not be relevant for arousal prediction. In this section, we explain the irrelevant feature pruning steps. Our first step is to remove all features with sparsity more than 0.99 as it helps in generalization of classification task and prevent overfitting.

We then use semantic relevance to remove more irrelevant features. News articles are categorized into various categories, such as entertainment, sports, politics, classifieds, etc. Users are recommended news articles based on the categories that they are interested in. An entity or topic popular in the sports domain may not be popular to users who are interested in some other domain, say politics or entertainment. Since the articles are ranked with respect to other articles in the same category, we use semantic relevance to prune features that are not relevant to the category.

For example, let's consider the following post from politics domain: Kellyanne Conway told CNN's Anderson Cooper that Donald J. Trump remains unconvinced that any breaches were part of an attempt to push him into the White House. Here, Anderson Cooper is a CNN journalist, who is one of the primary CNN anchor and author. Even though the term 'Anderson Cooper' is a noun and it appears very frequently in news posts, it has no connection with politics. Thus for news articles in the politics category, we identify terms that are not semantically related to politics and remove them from the list of potential features.

We use Google's Word2vec model [18] to measure semantic similarity. Word2vec model creates word embeddings by generating vector space from the text corpus where each word in the corpus is assigned to a vector in the space. We use the publicly available word2vec vectors that were trained on 100 billion words from Google News and each of these vectors has dimensionality of 300. We then find the similarity between word and category using word2vec similarity function. We select only those words that have similarity greater than 0.1.

3.4 Arousal Based Post Classification

In this paper, we use the post content to do arousal prediction. Using Eq. 3, we find arousal score of all the posts and then sort them in decreasing order of the arousal. We select the first k posts as posts of high arousal and last k posts as posts of low arousal based on arousal value (in this paper, we set k to 5,000). We then find the features from these two classes of posts as discussed in Sects. 3.2 and 3.3. We then train a binary classifier using these features. Moreover, our classification algorithm suffers from a class imbalanced problem due to different variety of posts assigned to the classes which leads to generate unequal number of features for the classes. This results in biased prediction and misleading accuracy. To make an accurate prediction, we apply SMOTE algorithm [15] which maintains class balanced training set for the classification.

Further, integrating multiple classification techniques usually produces more improved and accurate results than a single classification technique. Dietterich et al. [7] have also shown that combining multiple classifiers give better prediction compared to a single classifier. Therefore, we use ensemble based Voting Classifier [21] that weigh several individual classifiers and combine them in order to get a classifier that outperforms every one of them. The Voting Classifier implements ‘hard’ and ‘soft’ voting. We use hard voting, where we predict the final class label as the class label that has been predicted most frequently by the classifiers. In other words, the predicted final class label for a particular sample is the class label that is predicted by majority of the classifiers when they perform the classification task individually on the same sample. In our experiment, we use Random Forest, Decision Trees, k -Nearest Neighbours, and Extra Tree classifiers. As each classifier gives different accuracy while performing the classification on the same sample, we assign the different weights to these classifiers based on their prediction accuracy (described in Sect. 4.2). A new post that doesn’t have arousal is passed to the Voting Classifier which predicts whether the post would achieve high arousal or not.

3.5 Determining the Topics of High Arousal

We use posts with high arousal to find high arousal topics. The topical entities present in a post affect the arousal of the post [3]. For example, a post on president ‘Donald Trump’ is expected to generate higher arousal compared to posts on trivial topics.

In order to determine the topics of high arousal, we use the posts with high arousal. We perform Named Entity Recognition (NER) [16] over the posts and then use term-frequency to get topical entities that appear frequently in the posts. If two entities are consecutive, we merge them to form a single entity, as these consecutive entities often refer to a single entity, such as, Donald Trump, Hillary Clinton, Brad Pitt, etc. We also remove those unigrams that are part of some bigram and they refer to a single person, place or thing. For example, ‘Obama’ and ‘Barack Obama’ both represent the former president of United States. We remove the redundant term ‘Obama’ from the list of the topical entities. We do not remove unigrams that appear significantly without its superset bigram, such as gold, Olympics, President, Night, etc., which are the parts of gold medal, Olympics gold, President Obama, Saturday Night respectively.

Using only NER, we may lose many important topics that lead to high arousal, such as vote, surgery, competition, punishment, etc. Moreover, NER may give many entities that are not related to the domain of interest. We therefore apply word2vec similarity module to select the domain specific topics.

4 Evaluations

In this section, we first describe our dataset. We then compare the performance of proposed methods and present the topics of high arousal from various categories of the news articles.

4.1 Experimental Setup

We perform our experiments on Facebook pages of news channels. These pages are publicly accessible through the Facebook Graph API [26]. Our proposed technique can be used in all types of social media news channels. In this paper, we use the Facebook page of CNN¹, which is one of the most popular television based news channel.

Our dataset contains the following information: post, audience reaction namely likes, comments, and shares; link to the actual news article, and various page attributes such as organization name, post creation time, reaction time, etc. Audience react to posts through likes, comments, and shares. Each comment reactions has an opinion text with the count of likes for that comment. We consider all the posts from April 2012 to December 2016, which aggregates to 33,324 posts and 226.83 million reactions. News posts are classified into different categories such as politics, health, sports, entertainment, etc.

We do arousal analysis for each category separately, as the number of audience reaction varies greatly with category. Some categories, such as politics, sports, etc., get disproportionately high share of attention from users. We also performed basic text pre-processing such as removing of stop-words, stemming and lemmatization.

4.2 Effectiveness of Methods

As mentioned in Sect. 3.4, we use the Voting Classifier, which is a blend of multiple classifiers. Our voting classifier uses Random Forest, Extra Tree, Decision Tree, and K-Nearest Neighbour (KNN) classifiers. We assign weights to these classifiers based on their performances in our classification task. Random Forest and Extra Tree give the best performance followed by Decision Tree, which performs better than KNN.

To evaluate the importance of different feature sets and the proposed feature selection technique using word2vec, we compute the accuracy, precision, recall, F1 score for the following feature sets: (1) POS tagged features, (2) TF based features, (3) Intersection of POS tagged and TF features, and (4) Union of POS tagged and TF features. Both POS tag and TF generate a large number of features. By doing intersection we get a smaller feature set, which contains the common features from the two set. By taking union we get a much bigger feature dataset. Our aim is to find which of these feature sets give the best classification accuracy. For each of these cases, we consider both with and without feature selection. For feature selection, we use word2vec to prune features that are not relevant to the domain. In the following table, ‘w/o’ and ‘w/’ means ‘without’ and ‘with’ respectively.

As can be seen from Table 1, POS without feature selection has 76.8% classification accuracy. Applying feature selection with POS does not improve the accuracy. Instead, its accuracy decreases when the number of features decreases.

¹ <https://www.facebook.com/cnn/>.

POS tagging generates noun features that can capture useful entities such as, Donald Trump, Hillary Clinton, Brad Pitt, etc. If we apply feature selection then some of the less popular entity names are removed during feature selection, as they show very less similarity with a post category while using word2vec similarity function, which results in lower classification accuracy. On the other hand, for the TF based feature set the accuracy and F1 score increases by 3.3% and 5.9% respectively with the use of feature selection. Using term frequency we get many terms that are frequent but has no relevance to the post category. We are able to prune such irrelevant terms using word2vec.

Table 1. Performance evaluation of arousal prediction

Feature Set	Accuracy %	Precision %	Recall %	F1 score %
POS w/o FS	76.8	58.4	54.5	56.4
POS w/ FS	75.2	56.2	54.0	55.1
TF w/o FS	75.7	58.9	56.2	57.5
TF w/ FS	79.0	66.3	60.9	63.4
{POS \cap TF} w/o FS	68.4	47.6	48.2	47.9
{POS \cap TF} w/ FS	67.3	54.8	57.9	56.3
{POS \cup TF} w/o FS	77.0	60.2	54.3	57.0
{POS \cup TF} w FS	81.0	64.5	56.1	60.0

Intersection of POS and TF feature sets (w/ or w/o feature selection), gives the lowest classification accuracy. One of the reasons for this is that the intersection of POS and TF feature sets results in less number of features. Many important features that are frequent but not noun or noun but not frequent, are ignored while performing the intersection of two feature sets. Applying feature selection does not improve accuracy. On the contrary, it decreases the accuracy because it further reduces the number of features and ignores some of the frequent noun features.

It is interesting to note that taking the union of frequent and tagged feature sets without feature selection, improves the accuracy compared to the intersection of feature sets. The reason for this is that the union of both feature sets generate a sufficient number of frequent or tagged features which are able to capture the relevant entities. However, the best result (81%) comes when we apply feature selection on the union of POS, TF feature sets. One of the reasons for this is that word2vec method in feature selection generates word embeddings that preserve the aspect of the word’s context, which is an effective means to capture semantic relevance.

Overall, among the four feature set, the feature set obtained using union of POS and TF features gives the best classification, with 81.0% accuracy and 60.0% F1 score, and the one using intersection gives the worst classification, with 67.3% accuracy and 56.3% F1 score with feature selection, and 68.4% accuracy

and 47.9% F1 score without feature selection. The TF feature set with feature selection gives the highest F1 score of 63.4% compared to the second highest F1 score of 60% using union of POS and TF features.

4.3 Analyzing the Topics of High Arousal

We present some potential topics from four categories namely Politics, Health, Entertainment, and Sport that can increase the chances of getting high arousal on a post.

Table 2. Topics of High Arousal from Different Categories

Category	Topics of High Arousal
Politics	Barack Obama, Bernie, Bush, campaign, Clinton, debate, Donald Trump, election, Hillary Clinton, immigration, Marco Rubio, poll, president Obama, White House, Republican candidates, United States, vote
Sport	Competition, quarterback, FIFA, first olympic, gold medal, grand slam, legend, Leo Messi, match, medal, Michael Phelps, NBA, NFL, Olympic gold, phelps, punishment, relay, rugby, Ryan, Serena, soccer, superyacht, swimmers, tennis, Wimbledon, world series
Health	Autism, kids, blood, body, brain, cancer, children, conjoined twins, doctor, drug, Ebola, experts, hospital, life, listeria, lunch, measles, medical, outbreak, pain, prevent, recovery, risk, surgery, transformation, treatment, tumor, virus, world health
Entertainment	Actor, Angelina, Baldwin, Brad Pitt, breaking news, comedian, emotional, family, fan, favorite, films, Grammys, Harry Potter, history, Jennifer, Katy Perry, Lady Gaga, legend, Miss universe, NBC, night live, Saturday night, series, singer, talent, tv show, weekend

As can be seen in Table 2, all the categories show very prominent topics which are quite attractive and engaging. In the politics category, there are the topics such as Donald Trump, Barack Obama, Republican candidates, immigration, attacks, debate, etc., which can arouse people to interact or comment on posts related to these topics. Similarly, the topics captured under the entertainment category such as Brad Pitt, Katy Perry, Jennifer, Lady Gaga, comedy, movie, music, etc., are highly trending topics that can garner huge user response. These celebrities often have a huge eager fan base who actively respond to every news about them. Sport is another one of the most followed categories. News related to sport often covers major sport events and the players involved in these events. Sport events such as FIFA world cup, Olympics, Wimbledon and Michael Phelps the Olympic swimming champion are some of the appealing topics that can

attract the user attention. Health is a major concern throughout the globe. We can see that news on issues such as the Ebola outbreak, the life threatening diseases, symptoms and precaution measures for various diseases are critical.

5 Conclusion

In this paper, we formulated the problem of predicting the arousal of news articles on social media news pages. High arousal on a news content indicates that people are interested in interacting with the content by providing their opinions. We modeled it as a classification problem and predicted if a news article can achieve high arousal. We extracted the arousal determining features from the content of news articles. Using the best feature set for classification, we achieved overall accuracy of 81%. We further analyzed the features or topics of high arousal from predominant categories such as politics, sport, health, etc. Interestingly, we found that news articles related to some particular topics such as popular celebrity, event, or controversial topics achieved high arousal. On the other hand, news articles related to some general issues or topics did not gain much arousal.

References

1. Aggarwal, C.C.: Recommender Systems. Springer, Cham (2016). <https://doi.org/10.1007/978-3-319-29659-3>
2. Ahmed, M., Spagna, S., Huici, F., Niccolini, S.: A peek into the future: predicting the evolution of popularity in user generated content. In: WSDM, pp. 607–616. ACM (2013)
3. Bandari, R., Asur, S., Huberman, B.A.: The pulse of news in social media: forecasting popularity. In: ICWSM, vol. 12, pp. 26–33 (2012)
4. Bucher, T.: Want to be on the top? Algorithmic power and the threat of invisibility on Facebook. *New Media Soc.* **14**(7), 1164–1180 (2012)
5. Castillo, C., El-Haddad, M., Pfeffer, J., Stempeck, M.: Characterizing the life cycle of online news stories using social media reactions. In: Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing, pp. 211–223. ACM (2014)
6. Das, A.S., Datar, M., Garg, A., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: WWW, pp. 271–280. ACM (2007)
7. Dietterich, T.G.: Ensemble learning. In: The Handbook of Brain Theory and Neural Networks, vol. 2, pp. 110–125 (2002)
8. Esiyok, C., Kille, B., Jain, B.J., Hopfgartner, F., Albayrak, S.: Users' reading habits in online news portals. In: Proceedings of the 5th Information Interaction in Context Symposium, pp. 263–266. ACM (2014)
9. Figueiredo, F., Pinto, H., Belém, F., Almeida, J., Gonçalves, M., Fernandes, D., Moura, E.: Assessing the quality of textual features in social media. *Inf. Process. Manage.* **49**(1), 222–247 (2013)
10. Kim, C., Yang, S.U.: Like, comment, and share on facebook: how each behavior differs from the other. *Public Relat. Rev.* **43**(2), 441–449 (2017)

11. Lee, J.: The double-edged sword: the effects of journalists' social media activities on audience perceptions of journalists and their news products. *J. Comput.-Mediated Commun.* **20**(3), 312–329 (2015)
12. Lee, J.G., Moon, S., Salamatian, K.: An approach to model and predict the popularity of online contents with explanatory factors. In: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), vol. 1, pp. 623–630. IEEE (2010)
13. Lin, C., Xie, R., Guan, X., Li, L., Li, T.: Personalized news recommendation via implicit social experts. *Inf. Sci.* **254**, 1–18 (2014)
14. Liu, J., Dolan, P., Pedersen, E.R.: Personalized news recommendation based on click behavior. In: Proceedings of the 15th International Conference on Intelligent User Interfaces, pp. 31–40. ACM (2010)
15. Lusa, L., et al.: Smote for high-dimensional class-imbalanced data. *BMC Bioinformatics* **14**(1), 106 (2013)
16. Manning, C.: Information extraction and named entity recognition (2012)
17. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: ACL (System Demonstrations), pp. 55–60 (2014)
18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
19. Naveed, N., Gottron, T., Kunegis, J., Alhadi, A.C.: Bad news travel fast: a content-based analysis of interestingness on twitter. In: Proceedings of the 3rd International Web Science Conference, p. 8. ACM (2011)
20. Petrovic, S., Osborne, M., Lavrenko, V.: Rt to win! Predicting message propagation in twitter. *ICWSM* **11**, 586–589 (2011)
21. Rokach, L.: Ensemble-based classifiers. *Artif. Intell. Rev.* **33**(1), 1–39 (2010)
22. Suh, B., Hong, L., Pirolli, P., Chi, E.H.: Want to be retweeted? Large scale analytics on factors impacting retweet in Twitter network. In: 2010 IEEE Second International Conference on Social computing (socialcom), pp. 177–184. IEEE (2010)
23. Tatar, A., Antoniadis, P., De Amorim, M.D., Fdida, S.: From popularity prediction to ranking online news. *Soc. Network Anal. Min.* **4**(1), 1–12 (2014)
24. Tatar, A., Leguay, J., Antoniadis, P., Limbourg, A., de Amorim, M.D., Fdida, S.: Predicting the popularity of online articles based on user comments. In: Proceedings of the International Conference on Web Intelligence, Mining and Semantics, p. 67. ACM (2011)
25. Tsagkias, M., De Rijke, M., Weerkamp, W.: Linking online news and social media. In: WSDM, pp. 565–574. ACM (2011)
26. Weaver, J., Tarjan, P.: Facebook linked data via the graph API. *Semant. Web* **4**(3), 245–250 (2013)
27. Weng, L., Menczer, F., Ahn, Y.Y.: Predicting successful memes using network and community structure. In: ICWSM (2014)
28. Wu, S., Tan, C., Kleinberg, J.M., Macy, M.W.: Does bad news go away faster? In: ICWSM. Citeseer (2011)
29. Yano, T., Smith, N.A.: What's worthy of comment? Content and comment volume in political blogs. In: ICWSM (2010)
30. Zaman, T.R., Herbrich, R., Van Gael, J., Stern, D.: Predicting information spreading in Twitter. In: Workshop on Computational Social Science and the Wisdom of Crowds, Nips, vol. 104, pp. 17599–601. Citeseer (2010)

Sentiment Analysis of Tweets in Malayalam Using Long Short-Term Memory Units and Convolutional Neural Nets

S. Sachin Kumar^(✉), M. Anand Kumar, and K. P. Soman

Centre for Computational Engineering and Networking, Amrita University,
Coimbatore, India
sachinme@gmail.com

Abstract. Sentiment analysis in natural language processing (NLP) is an important task as the text content contains several opinions about events, product and movie reviews, trading, marketing etc. In the past decade, researchers have performed the sentiment analysis using hand-crafted features and machine learning methods such as support vector machines, naive Bayes, conditional random field, maximum entropy method etc. Sentiment analysis on social media text gained lot of popularity as it contain recommendations and suggestions. Compared to the high-resource languages such as English, Chinese, French etc., sentiment analysis task in low-resource language suffers due to (1) absence of annotated corpus, (2) tools to extract features. The present paper derives its motivation and addresses the sentiment analysis task for tweets in Malayalam, a low-resource language. Due to the absence of dataset, 12922 tweets in Malayalam language is collected and annotated to either of three sentiment categories namely positive, negative and neutral. Recently, deep learning methods like long short-term memory (LSTM) and convolution neural network (CNN) have gained popularity by showing promising results for the problems in speech and image processing, tasks in NLP via learning feature rich deep representation from the data automatically. The current paper is first in its attempt to perform sentiment analysis of tweets in Malayalam language using LSTM and CNN. The paper presents 10-fold cross-validation results.

1 Introduction

Sentiment Analysis is an important task in Natural Language Processing (NLP). The text data can contain information as opinions and/or sentiment about several events, products or brands, movies, services, etc. It also contains the information like emotions - happy, love, sad, anger, drool etc., suggestions and recommendations. Analyzing such text provides a comprehensive view about an opinion or sentiment which can be categorized into positive, negative and neutral. Online forums and discussion groups, chat services, blogs and microblogs, social networking sites such as facebook, google plus, twitter etc. are the virtual place where people share their opinion or express their sentiment. Several

such opinions spawns different other threads to exchange and interact on individual views (opinion). The existing technologies and applications facilitates for explosive rise of data in textual form every day [1,2]. As the text data contains several opinion and sentiment which gives a comprehensive perspective, it has become an interesting area of research to extract such information from texts automatically. The textual content has extensively become favorites for companies, government, researchers etc. Sentiment Analysis task is to extract the sentiment information, latent or hidden in the text content. Most of the text data in social media are informal in nature which makes the task of sentiment analysis more challenging. People express their view in social media sites using their native language or English which makes the extraction of sentiment or opinion from non-English text as another challenge.

In the past, the sentiment analysis task were performed using Naive Bayes, Support Vector Machines [3,4] etc. Recently, deep learning methods have gained huge popularity by showing impressive results in Speech Recognition, Computer Vision related tasks and tasks in Natural Language Processing [5–7]. Traditional methods when compared to deep learning methods have its limitation in analyzing the raw data or in its natural form. Whereas in deep learning, it learns the representation of data at multiple levels by combining simple nonlinear functions which transforms the raw data to a higher abstract level representation. During this transform operations, highly nonlinear functions are learned. In NLP, deep learning methods are used to learn vector representations for words via composition of learned vectors of words and neural language model, semantic role labeling, part-of-speech tagging (POS), named entity recognition (NER), paraphrase selection, sentiment analysis, Statistical Machine Translation (SMT), modeling sentence using Convolution Neural Network [8–16] etc.

The present paper takes the motivation to address the sentiment analysis of twitter data in Malayalam language. Malayalam is a language spoken predominantly by the people of Kerala, India [17]. Malayalam is a low-resource language in terms of tools for understanding natural language text with its applications and social media text. Hence, this paper derives its motivation as a first attempt to come up with deep learning based sentiment analysis for Malayalam twitter data. The dataset collected consists of 12922 manually annotated tweets in Malayalam language. Each tweet is labeled as positive, negative or neutral to denote its sentiment or polarity. Traditional methods are based on hand-crafted features. This process consumes lot of time and choosing proper feature is a concern. Inspired from the promising results obtained using deep learning in NLP tasks, the current work proposes deep learning based sentiment analysis. The experiments in the current paper are performed using deep learning methods - (1) long short-term memory (LSTM) and (2) convolutional neural network (CNN) and evaluated using metrics - f1-score, precision, recall and accuracy. The rest of the paper is organized as, Sect. 2 gives an outline on the articles related to deep learning, sentiment analysis etc. Section 3 describes about the deep learning methods used for performing experiment. Section 4 discusses the

proposed approach used to tackle sentiment analysis on Malayalam twitter data using deep learning method. The experiment and results are discussed in Sect. 5.

2 Related Work

Analyzing the sentiment of text data gives an overview of peoples or users opinion or views or emotions related to any event, product or brands, movies etc. Several research work proposed sentiment analysis via linguistic and machine learning based methods in the past. Recent advancement in deep learning based methods motivated and fasten the research to explore the sentiment analysis in English and non-English textual contents. Traditional machine learning methods require hand-crafted features which consumes lot of time. The state-of-the-art result for polarity detection using Support Vector Machines was shown in [18]. The authors detected sentiment at message-level (SMS, tweets) and term-level (words). The features used consists of character n-grams, word n-grams, POS, hastags, capitals in sentence, negated context etc. Their approach obtained an f1-score of 0.8893 for term-level and 0.6902 for message-level task. Sentiment analysis was performed at word, phrase, sentence and document level. In [19], the authors proposed a sentiment analysis using features such as unigram, bigram, unigram and bigrams, pos and evaluated using naive bayes, maximum entropy and svm. The authors showed that three classifiers performed similarly in their experiments. In [20], authors casted sentiment analysis problem as binary classification to identify positive and negative tweets, and three-way classification to identify positive, negative and neutral tweets. Three models were created using unigrams, 100 features and tree kernel. The authors also evaluated by combining the models which was found to be effective. The paper [21] discus the method to detect the sentiment at phrase-level. Features used consists of lexical scores as priors from dictionary and wordnet, n-gram calculated on syntactic constituent in phrase, and polarity score of the phrase and Logistic classifier was learned for detecting polarity in two cases: (1) positive, negative and neutral, (2) positive and negative. [22] authors presented sentiment at phrase-level by determining the phrase as neutral or polar initially, and then detects the polarity as positive or negative. The proposed approach tries to find the contextual polarity of phrases where a word in different context can take different polarity. For learning classifier, features used consists of word tokens, pos, context, prior information about polarity of words, composition of features etc. Sentiment analysis has also been applied at document-level. In [23], authors discusses unsupervised way of sentiment analysis via three-step procedure: (1) extract phrase which contains adverbs and adjectives by applying suitable pos tagger, (2) find phrase-level semantic orientation, (3) calculate the sentiment based on the average of semantic orientation. The experiments are performed on movie reviews and obtained an average accuracy of 74%.

Compared to the traditional methods, deep learning avoids the intensive task of feature engineering by learning features from the raw data itself. In traditional methods, the feature selection must be proper in order to handle the

order of words, syntactic and semantic structure among the words. Whereas deep learning networks inherently takes care of order of words. The vector representation of words found using deep learning methods captures the semantic and syntactic structure. In [24], the author proposed a recursive neural tensor net (RNTN) for predicting the sentiment at sentence and phrase-level. The proposed methods generates word vectors based on the semantic compositionality among word sequences. The approach has obtained an accuracy of 85.4% at sentence-level and 80.7% at phrase-level. Convolutional neural modelling of sentence was proposed in [25]. The method outperformed the evaluation obtained using ngram features on tweets and movie reviews. The convolutional network learns the compositional relations among words in a sentence thereby avoiding the use of any prior knowledge. Attempts are made to use the pre-trained models to find the sentiment of a text. However, the word-vectors learned lacks the knowledge of sentiment. [26] presents method to learn sentiment specific word vectors or embeddings. In order to learn sentiment oriented word embedding, authors used benchmarked dataset from Semeval 13. The authors extended the existing word vector model [11] to derive three neural nets for learning sentiment specific embedding. Sentiment specific word embedding learns the context and sentiment related information so that words that appear in similar context with different sentiment can be distinguished. In [27], authors proposed a deep convolutional model for tweets. The experiments were perform on dataset from Semeval 15 with 2 fully connected layers. The evaluation obtained an f1-score of 64.85%.

Though major research work in sentiment analysis is carried out in languages such as English, German, Chinese, Spanish etc., immense research is going on for sentiment analysis in Indian languages such as Hindi, Bengali, Punjabi, Marathi, Tamil, Telugu, Malayalam etc. In [28], shows the summary of the sentiment analysis shared task for tweets in Hindi, Bengali and Tamil. The shared task was the first attempt in this direction and the task was to detect the sentiment of tweets in Hindi, Bengali and Tamil. The maximum accuracy achieved for tweets in Hindi, Bengali and Tamil 55.67%, 43.2%, and 39.28%. The paper [29] presents a hybrid approach using deep learning for sentiment analysis in Hindi. The method proposes to use embedded word vectors learned via convolutional neural modelling with optimized features chosen through multi objective optimization. In [30], authors proposed naive bayes based sentiment classification on tweets in Hindi, Bengali and Tamil language [28]. The features are in binary form to show the presence of hashtag, emoticon, punctuations, urls etc. The proposed method gave highest score in the shared task in Indian language [28]. In [31], proposed a sentiment analysis approach using word sense disambiguation. An edge based method for finding similarity was used to compute the semantic distance among synsets. SVM was used to learn the classifier and obtained an f1-score of 0.74.

The current paper presents CNN and LSTM based sentiment analysis of tweets in Malayalam language. Due to the absence of dataset, tweets were extracted and manually annotated based on its sentiment as positive negative

or neutral. The CNN architecture followed for experiments have one convolution layer, a pooling layer and a softmax layer. For experimental purpose, four different filter sizes such as 32, 64, 128, 256 are used with stride as 1. The LSTM architecture used consists of 2 layers with four different cell configurations such as 32, 64, 128, 256. Dropout is used for both experiments to avoid the over-fitting issue. The optimization algorithm used is adam for both experiments. The architectures followed for the experiments using CNN and LSTM are similar to the one discussed in [16,42].

3 Background

Currently, deep learning approach in machine learning is the trend. It has the advantage of learning rich feature representation from data which avoids the feature selection process in traditional machine learning methods. Deep learning methods such as LSTM and CNN has shown promising results in speech and image processing, and tasks in NLP. In deep learning approach, the several layered architecture (deep layers) learns feature representation via back-propagation [10]. In general, two types of neural network architectures followed in language related tasks are (1) FF or Feed-forward (context is fixed, network connections will not occur in cycles) [14], (2) Recurrent Neural Network (length of context is not fixed, network connections are in loops) [32]. In FF method, the context length being fixed makes it deficient to capture contextual information when the length vary. This motivated to use RNN and its variants for NLP related tasks as it can capture context of arbitrary lengths (even all previous words). In this paper, CNN and LSTM are experimentally explored to use for sentiment analysis task of tweets in Malayalam language. The present paper uses convolution 1D as it has been widely used for NLP tasks like question-answering, information retrieval, sentence modelling, document modelling etc. Training RNN through back-propagation to capture the long-dependencies, the gradients can explode or decays/vanishes [33,34].

3.1 Recurrent Neural Network or RNN

RNN is an extension of traditional feed-forward neural nets. RNN is capable of handling sequences with varying length via hidden recurrent states connected in a loop (connected to themselves for feedback). This can be viewed as a passage through time and the activations at each time-step depends on the previous. For a given sequence $x_1, x_2, x_3 \dots x_n$, the RNN updates the hidden state h_t using the previous hidden state h_{t-1} and current input x_t as

$$h_t = f(Ax_t + Wh_{t-1}) \quad (1)$$

where f is a nonlinear function (element-wise operation) such as hyperbolic tangent, sigmoid, rectified linear unit function, exponential linear units [35], scaled-exponential liner units [36] etc. A, W are the learned input and recurrent weight matrices. During training RNN cannot capture long-term dependencies

(longer past observations) [37] as the magnitude of gradient shoots (or explodes) or decays (or vanishes) over the time. The unfolded RNN over the time become a deep FF net which suffers the decay/vanishing of magnitudes of gradients. The popular approaches to overcome this issue are (1) gradient clipping method [38], (2) Hessian-free optimization [39], (3) Long short-term memory (LSTM) [40]. In this paper, LSTM is used to train the model to detect the sentiment of tweets in Malayalam.

3.2 Long Short-Term Memory or LSTM

LSTM is a variant of RNN which contain memory-units or memory cell in addition to the normal recurrent unit. The memory-units are used to transfer information from longer past observations with the help of gate's which provides selected interaction with the hidden states to model the long-term dependencies. LSTM introduced three gating mechanisms such as input, forget and output gates and memory unit c_t to handle the long-term dependencies. The memory unit forms the core idea behind LSTM that can maintain its state over time using nonlinear gating operations to control the flow of information in or out of the unit. The memory unit accumulates the state information. Enabling the input gate i_t allows the information to get accumulated in the memory unit or cell when new input comes. During this process, the previous cell state information c_{t-1} can be avoided when the forget gate f_t is enabled. To propagate the information in cell state c_t to the final state h_t , the output gate is enabled. The basic formulas of LSTM at t^{th} time-step are (input to the LSTM are x_t , h_t and c_{t-1})

$$i_t = \sigma(A^i x_t + W^i h_{t-1}) \quad (2)$$

$$fg_t = \sigma(A^{fg} x_t + W^{fg} h_{t-1}) \quad (3)$$

$$o_t = \sigma(A^o x_t + W^o h_{t-1}) \quad (4)$$

$$\hat{c}_t = \tanh(A^c x_t + W^c h_{t-1}) \quad (5)$$

$$c_t = \hat{c}_t \odot i_t + c_{t-1} \odot fg_t \quad (6)$$

$$h_t = o_t \odot \tanh(c_t) \quad (7)$$

where i, fg, o, c, \hat{c} are the input, forget, output gate, cell state, intermediate cell state and $A^i, W^i, A^{fg}, W^{fg}, A^o, W^o$ are the learned weight matrices. The operations $\tanh(\cdot)$ and $\sigma(\cdot)$ are performing element-wise tangent and sigmoid operations, and \odot denotes the element-wise multiplication operation. Compared to the traditional recurrent net which updates the content at every time-step, LSTM net with the help of gates and memory unit (or cell) is able to carry useful information (important feature) detected in the early stages over long distance enabling the capturing of long-term dependencies.

3.3 Convolutional Neural Network or CNN

CNN is a popular deep learning approach used in vision related problems and recently it gained popularity in addressing tasks in NLP. The idea of CNN is adopted from the convolution operation in time-domain. Consider a discrete function of the form $f(x) \in \mathbb{R}^l$ and another discrete function or kernel function $g(x) \in \mathbb{R}^d$, then the convolution operation is defined as

$$h(y) = \sum_1^d f(x) \cdot g(y \cdot d - x + c) \quad (8)$$

where $c = d - s + 1$ is an offset and s is the stride. This operation is adopted in deep learning and the kernel function or filters are learned from the data. In the current work, 1-dimensional CNN is used. Let $D = \{c_1, c_2, \dots, c_l\}$ denote the sequence of tokens and l denotes the no of words in the sequence. V denotes the vocabulary of words, d is the dimensionality of embedding for words, $V_D \in \mathbb{R}^{d \times l}$. The objective is to perform 1-dimensional convolution and learn filters (or kernel functions) specific to the data $f \in \mathbb{R}^{d \times p}$ where p is the window of characters to form feature map f_m over the window of tokens $V[* , k : k + p]$. The feature map thus obtained is operated by activation function, e.g. rectified linear units (ReLU) $f_m(x) = \max(0, x)$. The non-linear function allows to learn a decision boundary with non-linear nature. After obtaining feature maps, pooling operation is performed to down sample the size of feature map. The most commonly used pooling approaches are average and max pooling. In the current work max pooling is used which returns the maximum value by operating on the feature maps. The window size used for pooling is 2. CNN approach learns the features by itself, eliminating the difficult task of finding the right features. CNN efficiently exploits the locality information via convolution operation applied on to the input vector (one-hot, vector indices etc.). In this paper, the tokens are mapped to their corresponding indices in the V . The output of the pooling operation is passed to a fully connected softmax layer. This layer computes the probability distribution which gives information about the labels.

$$\text{soft max}(x^T w + b) = \frac{\exp(x^T w_i + b_i)}{\sum_{j=1}^J \exp(x^T w_j + b_j)} \quad (9)$$

where w_i, b_i corresponds to the weight vector, bias of j^{th} class.

3.4 Regularization via Dropout

In order to handle the over-fitting issue of parameters (cause for poor models), regularization via dropout is utilized [41] in the current work for LSTM and CNN. The dropout parameter is fixed as 0.3 for the current experiment. Dropout selects few neurons randomly and removes (or drops connections input and outgoing to neurons) during the training process such that the activations from

those during forward and backward pass to find weights are not used. Let $l \in \{1, 2, \dots, L\}$ denotes L hidden layers, y^l denote the output vector from layer l , W^l and b^l denotes the weight and bias at layer l . In the dropout operation, the sampled output vector \hat{y}^l is obtained as

$$\hat{y}^l = r^l * y^l \quad (10)$$

where r^l is a vector of random variables following Bernoulli distribution and the update will be

$$y^{l+1} = W^{l+1}y^l + b^l \quad (11)$$

The vector r^l is element-wise multiplied with y^l to obtain the thinned or sampled new output vector. Intuitively, during the dropout process, the weights of certain neurons wont get tuned for some specific features or specializes. To balance the missing neurons, neighboring neurons will be used to find the representation during training and co-adaptions will not happen as in the case of normal network. This will make the network capable to handle the over-fitting issue.

3.5 Training and Softmax Classifier

The values of the parameters are fixed by defining a cost function and minimizing the error of the cost function (w.r.t to parameters) using gradient descent method. Through gradient descent algorithm and back-propagation, the weights are adjusted as the gradient of the error, $-\frac{\partial E}{\partial w}$, of the defined cost function decreases. This means that if the weight parameter is perturbed, how much it effects error. The negative sign is added as the direction where the error is reducing. The learning happens by propagating the gradients back from the error. Properly learned weights provide better classification. For the sequence classification problem, loss function commonly used is cross-entropy loss. It is defined as

$$H_{y'}(y) := - \sum_i y'_i \log(y_i) \quad (12)$$

where y_i denotes the probability distribution of predicted i^{th} class and y'_i denotes the true probability distribution. The loss function finds the importance as it gives a measure about data loss, which can be obtained using predicted and true labels. During testing, the output layer of the aforementioned network is given to Eq. 9 which provides the probability distribution related to each class.

4 Experiment and Evaluations

The sentiment analysis of tweets in Malayalam language are performed using CNN and LSTM. A 10-fold cross-validation results are obtained and an average score is obtained using metrics such as precision, recall, F1-score and accuracy. The experiments tries to find out (1) how well the CNN and LSTM models performs through the evaluations at different parameter configuration (2) and to

find the configuration for CNN and LSTM which gives better results for identifying the sentiment of tweets in Malayalam language. The evaluations give a comparison between the performance obtained for CNN and LSTM models at different parameter configurations. All the experiments are ran for 100 epochs to generate the model and 10-fold cross-validation is performed. The experiments are performed at word-level. Initially, a unique dictionary was created from the annotated dataset and to find the vector representation for each tweet, after pre-processing (removing urls, @-mentions, #-tags, unwanted recurring symbols) each token in the sequence is replaced with its index location in the dictionary thus forming a vector representation for the tweet. The vectors corresponding to each tweet are padded with zeros to make the length same.

4.1 Dataset

Due to the lack of dataset for experiment, 12922 tweets in Malayalam language were extracted using the API and manually annotated each one into three sentiment category or class such as positive, negative and neutral. Table 1 gives the detailed distribution of sentiment data used for experiments.

Table 1. Sentiment data

Positive	3170
Negative	3126
Neutral	6626
Total	12922

4.2 Models

Initially, a unique vocabulary of token is created with which every tweets corresponding indices vector is generated. The tokens in each tweet are replaced with its corresponding index number in the vocabulary. Zeros are padded to the vectors to make the length same. In this experiment the length of such vector is fixed as 100 and it is fed to the network. For training with CNN using adam optimizer, four different models are generated by keeping the number of filters in each case as 32, 64, 128, 256. The network consists of one convolution layer with global max pooling and stride chosen as 1. For CNN based experiments, the architecture used in the current work closely follows [16]. One of the aim of the experiment is to find the filters from the set suitable for the sentiment analysis task. Similarly, for LSTM, to find the model which gives better results, four different models are generated by changing the number of LSTM cells as 32, 64, 128, 256. The architecture followed for LSTM is similar to one discussed in [42]. In order to deal with over-fitting issues, dropout parameter is set.

The learning-rate parameter is set to take default value. Three different activation functions are chosen during evaluations as shown in Eqs. 13, 14 and 15. The parameter α is set to default values for each functions depending on which the negative values get skewed.

1. Rectified Linear Units (ReLU)

$$f_m(x) = \max(0, x) \quad (13)$$

2. Exponential Linear Units (ELU) [35]

$$f(x) = \begin{cases} x, & x > 0 \\ \alpha(\exp(x) - 1), & x \leq 0 \end{cases} \quad (14)$$

3. Scaled Exponential Linear Units (SELU) [36]

$$f(x) = \lambda \begin{cases} x, & x > 0 \\ \alpha(\exp(x) - 1), & x \leq 0 \end{cases} \quad (15)$$

4.3 Result and Discussion

Tables 2 and 3 exhibits the evaluation results obtained on the prepared dataset using CNN and LSTM models. The number of filters chosen for CNN based experiments are 32, 64, 128, 256. The kernel size is set as 3 for all the cases. The filters are learned through back-propagation process driven from the data. The filter values acts as the weight values for CNN and the purpose is to learn relevant patterns or specific features from the data. For 32 filter operations, it creates 32 feature different feature maps which undergoes pooling operation before its made fully connected and the dense vector is passed to the softmax layer. Similarly, with filters 64, 128, 256. The non-linearity is introduced using the chosen activation functions (ReLU, ELU, SELU in the present paper). In order to deal with the over-fitting issue, dropout parameter is set to 0.3 which randomly removes 30% connections. This ensures that neurons learns co-adapted features and not get tuned to learn specific features. Since the dataset prepared for the experiment need more number of tweets to generalize the results, the paper presents an evaluation obtained via 10-fold cross-validation and Table 2 shows the average scores obtained during the evaluation. For the iterative training of the network with adam optimizer, the epoch parameter is set to 100, learning-rate as 0.01, mini-batch set as 64. It is observed that as the epochs increased, the loss was decreasing during training. The present work is first in this direction to perform sentiment analysis on tweets in Malayalam language using CNN. From the evaluation results, it can be observed that as the number of filters increases the scores are improving. This behavior is similar for the three activation functions. ELU and SELU activation functions, compared to ReLU, improves the learning capacity of the deep networks which is evident from the results obtained. The α parameter chosen for ELU is default (1.0). SELU activation function is a variant which improves ELU and the effect of ELU, SELU

on improving the learning characteristics of the deep network is well explored in [35, 36]. The experiment conducted using the three activation functions shows its effect and importance for using it to train the network for tasks like sentiment analysis. Table 3 shows the results obtained for LSTM. The number of LSTM units or the hidden states chosen for the experiments are 32, 64, 128, 256 and each network is trained for 100 epochs using adam optimizer with learning-rate as 0.01, mini-batch size set as 64. It is observed that as the epochs increased, the loss was decreasing during training. Similar to CNN, to introduce nonlinearity in the experiments using LSTM, ReLU, ELU, and SELU activation functions are used. Observing the cross-validation results, ELU and SELU, compared to ReLU, improves the scores as the number of LSTM unit increases. The challenge and the motivation to work on sentiment data (e.g. tweets) in Malayalam language can be observed from the following examples (the tweets are converted to phonemic form for general understanding, originally its in Malayalam language).

Table 2. Evaluations of CNN models

Activation Function	Method	Precision	Recall	F1-score	Accuracy
ReLU	CNN_32	0.9546	0.9534	0.9554	0.9534
	CNN_64	0.9664	0.9605	0.9620	0.9605
	CNN_128	0.9743	0.9729	0.9733	0.9729
	CNN_256	0.9759	0.9746	0.9750	0.9746
ELU	CNN_32	0.9664	0.9632	0.9673	0.9632
	CNN_64	0.9743	0.9721	0.9732	0.9721
	CNN_128	0.9768	0.9753	0.9757	0.9753
	CNN_256	0.9825	0.9819	0.9821	0.9819
SELU	CNN_32	0.9601	0.9693	0.9695	0.9693
	CNN_64	0.9769	0.9753	0.9757	0.9753
	CNN_128	0.9779	0.9766	0.9769	0.9766
	CNN_256	0.9794	0.9784	0.9787	0.9784

e.g.; ayyO... ivan lApuTOpu mOshTiCCatu kaNTO..?? kaLLan ANEngkilum sammatiCCu mOne!

eg; ishTappeTTAl...oru...laikku...I...paTaththinum...I...pEjinum

eg; viTTile kASu aTiCCumARRi nATu viTTallo

eg; en svaram pUviTum gAnamE.. I vINayil nI anupallavi..

From the dataset, it is observed that the tweets exhibit sentiments directly, indirectly as in sarcasm, proverbs, portions of songs etc. The advantage of deep learning is that it can learn by its own from more labeled examples which in-turn takes away the laborious task of feature engineering as in traditional methods. The experimental results obtained for CNN and LSTM shows that both provides

competing scores, event though LSTM with SELU activation function is better for the task of detecting sentiment of tweets in Malayalam language. The present work did the following

1. performs sentiment analysis task for tweets in Malayalam language on manually annotated dataset due to its absence.
2. addresses the sentiment analysis task via deep learning methods such as CNN and LSTM.
3. explores the effect of activation functions such as ReLU, ELU and SELU for the task and provided cross-validation results to support it.

Table 3. Evaluations of LSTM models

Activation Function	Method	Precision	Recall	F1-score	Accuracy
ReLU	LSTM_32	0.9685	0.9644	0.9655	0.9644
	LSTM_64	0.9700	0.9676	0.9682	0.9676
	LSTM_128	0.9764	0.9752	0.9755	0.9752
	LSTM_256	0.9827	0.9826	0.9828	0.9826
ELU	LSTM_32	0.9733	0.9710	0.9716	0.9710
	LSTM_64	0.9736	0.9719	0.9723	0.9719
	LSTM_128	0.9748	0.9733	0.9737	0.9733
	LSTM_256	0.9775	0.9763	0.9766	0.9763
SELU	LSTM_32	0.9726	0.9711	0.9714	0.9711
	LSTM_64	0.9730	0.9704	0.9711	0.9704
	LSTM_128	0.9738	0.9719	0.9725	0.9719
	LSTM_256	0.9823	0.9824	0.9823	0.9824

5 Conclusion and Future Work

The paper presents sentiment analysis of tweets in Malayalam language using deep learning approaches such as convolutional neural network (CNN), and long short-term memory units (LSTM). Due to the absence of dataset, tweets in Malayalam language are extracted using API and manually annotated 12922 tweets into three sentiment categories such as positive, negative and neutral. The current work is first in its kind in this direction. In the experiment, the CNN is trained using four different filters taken as 32, 64, 128, 256. Since the dataset prepared for the experiment need more number of tweets to generalize the results, the paper presents an evaluation obtained via 10-fold cross-validation. For the experiment using LSTM, four different LSTM cell parameters such as 32, 64, 128, 256 are considered. Both the experiment uses three different activation functions such as ReLU, ELU and SELU to introduce nonlinearity in the

network. It is observed from the experiments that activation functions ELU and SELU improves the scores for CNN and LSTM. Comparing the results obtained, LSTM with SELU activation function is having competing results than CNN with 0.9823 as F1-score, 0.9824 as recall, 0.9823 as precision and 0.9824 as accuracy. As future work, the proposed approaches are planned to apply on a larger corpus.

References

1. <http://wikibon.org/blog/big-data-statistics/>. Accessed 25 July 2017
2. <http://www.scidev.net/global/data/feature/big-data-for-development-facts-and-figures.html>. Accessed 25 July 2017
3. Mullen, T., Collier N.: Sentiment analysis using support vector machines with diverse information sources. In: EMNLP, pp. 412–418 (2004)
4. Sarah S.: Machine Learning Approaches to Sentiment Analysis Using the Dutch Netlog Corpus, CTRS-001 (2009)
5. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS, pp. 1106–1114 (2012)
6. Graves, A., Mohamed, A.-R., Hinton, G.: Speech recognition with deep recurrent neural networks. In: ICASSP (2013)
7. Chorowski, J., Bahdanau, D., Cho, K., Bengio, Y.: End-to-end continuous speech recognition using attention-based recurrent NN: first results, arXiv preprint [arXiv:1412.1602](https://arxiv.org/abs/1412.1602) (2014)
8. Pennington, J., Socher, R., Manning, C.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543. Association for Computational Linguistics, Doha, October 2014
9. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) NIPS, pp. 3111–3119 (2013)
10. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **12**, 2493–2537 (2011)
11. Socher, R., Huang, E.H., Pennington, J., Andrew, Y. Ng., Manning, C.D.: Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In: NIPS, vol. 24 (2011)
12. Cho, K.: Learning phrase representations using RNN encoder-decoder. In: EMNLP, pp. 1724–1734 (2014)
13. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. *JMLR* **12**, 2493–2537 (2011)
14. Bengio, Y., Ducharme, R., Vincent, P.: A neural probabilistic language model. *J. Mach. Learn. Res.* **3**, 1137–1155 (2003)
15. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A Convolutional neural network for modelling sentences. In: ACL (2014)
16. Yoon, K.: Convolutional neural networks for sentence classification. arXiv preprint [arXiv:1408.5882](https://arxiv.org/abs/1408.5882), (2014)
17. <https://en.wikipedia.org/wiki/Malayalam>. Accessed 30 July 2017

18. Mohammad, S., Kiritchenko, S., Zhu, X.: NRC-Canada: building the state-of-the-art in sentiment analysis of Tweets. In: SemEval, Georgia (2013)
19. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, pp. 1–12 (2009)
20. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of Twitter data. In: Proceedings of the Workshop on Languages in Social Media, LSM 2011, pp. 30–38. ACL (2011)
21. Agarwal, A., Biadys, F., Mckeown, K.: Contextual phrase-level polarity analysis using lexical affect scoring and syntactic n-grams. In: EACL, pp 24–32 (2009)
22. Wilson, T., Wiebe, J., Hoffman, P.: Recognizing contextual polarity in phrase level sentiment analysis. In: ACL
23. Turney, P.: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: ACL (2002)
24. Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: EMNLP, pp. 1631–1642. Citeseer (2013)
25. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences, arXiv preprint (2014)
26. Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., Qin, B.: Learning sentiment-specific word embedding for twitter sentiment classification, ACL, Long Papers, vol. 1, pp. 1555–1565. Association for Computational Linguistics (2014)
27. Stojanovski, D., Strezoski, G., Madjarov, G., Dimitrovski, I.: Twitter sentiment analysis using deep convolutional neural network. In: Onieva, E., Santos, I., Osaba, E., Quintián, H., Corchado, E. (eds.) HAIS 2015. LNCS (LNAI), vol. 9121, pp. 726–737. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19644-2_60
28. Patra, B.G., Das, D., Das, A., Prasath, R.: Shared task on Sentiment Analysis in Indian Languages (SAIL) Tweets - an overview. In: Prasath, R., Vuppala, A.K., Kathirvalavakumar, T. (eds.) MIKE 2015. LNCS (LNAI), vol. 9468, pp. 650–655. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26832-3_61
29. Akhtar, M.S., Kumar, A., Ekbal, A., Bhattacharyya, P.: A hybrid deep learning architecture for sentiment analysis. In: Coling (2016)
30. Se, S., Vinayakumar, R., Anand Kumar, M., Soman, K.P.: AMRITA-CEN@SAIL2015: sentiment analysis in Indian languages. In: Prasath, R., Vuppala, A.K., Kathirvalavakumar, T. (eds.) MIKE 2015. LNCS (LNAI), vol. 9468, pp. 703–710. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26832-3_67
31. Kausikaa, N., Uma, V.: Sentiment analysis of English and Tamil Tweets using path length similarity based word sense disambiguation. IOSR-JCE **PP**, 82–89 (2016)
32. Mikolov, T., Karafiat, M., Burget, L., Cernock, J., Khudanpur, S.: Recurrent neural network based language model. In: Interspeech (2010)
33. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Networks **5**, 157–166 (1994)
34. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**, 533–536 (1986)
35. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (ELUs). In: ICLR (2016)
36. Klambauer, G., Unterthiner, T., Mayr, A., Self-normalizing neural networks, [arXiv:1706.02515](https://arxiv.org/abs/1706.02515) (2017)
37. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Networks **5**, 157–166 (1994)
38. Bengio, Y., BoulangerLewandowski, N., Pascanu, R.: Advances in optimizing recurrent networks. In: ICASSP (2013)

39. Martens, J.: Deep learning via Hessian-free optimization. In: ICML 2010, pp. 735–742 (2010)
40. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997)
41. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *JMLR* **15**, 1929–1958 (2014)
42. Chung, J., Gulcehre, C., Chao, K., Bengio, Y.: Empirical evaluation of gated recurrent networks on sequence modeling, [arXiv:1412.3555](https://arxiv.org/abs/1412.3555) (2014)

Improved Community Interaction Through Context Based Citation Analysis

Baishali Saha, Tanushree Anand, Anurag Sharma, and Bibhas Ghoshal(✉)

Department of Information Technology, Indian Institute of Information Technology,
Allahabad, Allahabad 211012, India

baishalisaha41@gmail.com, tanushreeanand100@gmail.com,
anuragiitald@gmail.com, bibhas.ghoshal@iitita.ac.in

Abstract. Traditional citation networks which form the basis of study of community interaction tend to leave out a lot of articles which are related to a community but have not been directly cited by the members of it. As a result, the parameters estimated during the study of community interaction remain fairly inaccurate. In this work, we tend to perform a more accurate community interaction study by proposing a context-aware citation network which allows inclusion of papers to a community which have both direct as well as indirect relevance to the existing members of the community. A comparative analysis of computer science community networks built upon the proposed citation network and traditional citation network using the CiteSeer dataset show about 20–30% better results in favour of the former.

1 Introduction

Since the last decade, citation network has emerged as the most promising way to organize scientometric data in order to draw inferences about publications, authors and their interactions. It has also proved to be effective in studying the community structure in different fields such as physics [1] and computer science [2]. However, using the regular citation network for community analysis has some limitations. A lot of articles which are related to a community but have not been directly cited by the members of it are left out. Consider for example, the paper on ‘*A binary feedback scheme for congestion avoidance in computer networks (1990)*’ has not been cited by ‘*Myths About Congestion Management in High-Speed Networks (1992)*’. However, the former has an influence on the research work of the later and should have an edge between the two in the citation network. Unfortunately, the regular citation network has no means to uncover these indirect citations, hence the community interaction study remains inaccurate.

Such kind of related papers can be uncovered by considering the citation contexts of the paper which provides a useful way to identify the main contributions of a scientific publication. Authors refer to the articles by briefly presenting key points of the cited article in citation context and thus citation context contains

representative keywords for the cited work. These keywords provide a broader description of the cited paper, thereby helping in inclusion of papers which have impact on the cited paper but may have been left out in traditional citation network.

In this work we propose to build a context-based citation network [3] as an improvement to the traditional citation network and show via comparisons that the study of community interaction becomes more accurate if performed on context-based citation network. The comparative analysis was performed using the same set of metrics in both cases and it was found that the context-sensitive approach shows a 20–30% increase in the FOMD and Flake-ODF values while rest of the metrics have shown around 5–7% increment in their values.

1.1 Related Work

A citation context is essentially the text surrounding the reference markers used to refer to other scientific works as shown in Fig. 1. Citation context have been for recommending high quality citations using neural probabilistic model as proposed in [3, 4]. However, citation context can have other implications as well, such as providing necessary inputs to improve the traditional citation network which is the main motivation for our work. In [5], structural and functional definitions of network communities are distinguished.

Note that segmentation has two major uses. It may be performed in order to determine when the underlying model that created the time series has changed [19, 20], or segmentation may simply be performed to create a high level representation of the time series that supports indexing, clustering and classification [20, 30, 31, 37, 39, 42, 44, 46, 48, 52, 57].

Fig. 1. Highlighted citation context of papers referred by 19 & 20

1.2 Contribution

The prime focus of this work has been to improve the traditional citation network, by performing citation context analysis, so as to include more papers that are similar/relevant to a particular paper but are not directly cited by the paper itself. To achieve this we have modified the directed citation network to become a term labelled directed citation network based on which a similarity index has been calculated for every pair of non adjacent nodes and thereby include such an edge if it crosses a certain threshold (Sect. 3). This is followed by building of communities using the context-sensitive citation network, analyzing its outcome and comparing with the traditional citation network.

2 Dataset Description

The dataset considered in this work is from the Citeseerx digital library¹ consisting of 63158 unique papers. The data was scraped and saved in a database, the schema of which is shown in Fig. 2.

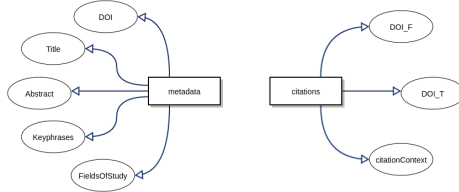


Fig. 2. ER diagram of the dataset

2.1 Field Tagging

The field tagging process assigns fields of study to each article in our filtered dataset. One particular paper can be interdisciplinary and can belong to multiple communities in the network hence has multiple fields of study.

Our dataset did not inherently have the required field information hence we used the Microsoft Academic Search Engine for field tagging (around 55% of papers) and the remaining untagged papers have been tagged by us using their titles (around 11.11%). The papers of computer science domain are categorized into twenty-three Computer Science Fields of Ground-truth Communities as noted in Table 1. As a result, out of all the articles, around 43000 papers get divided into clusters of communities.

2.2 Classification of Papers into Single Communities

To classify papers tagged by Microsoft Academic Database in multiple communities to a single community we perform Citation-enhanced Keyphrase Extraction similar to [6] and generate phrases with inverse document frequency (idf) values. We then use these idf values to score papers for various communities and classify the paper into one with the highest score.

Let there be a research paper, 'P', and some communities $C := \{C_1, C_2, \dots, C_n\}$. And there be a function, $f(P) \in C$. Now suppose papers which cite P have contexts: $S = \{S_1, S_2, \dots, S_j\}$. We follow the following procedure:

¹ <http://citeseerx.ist.psu.edu/index>.

Table 1. Computer science ground truth communities

	Community	Abbreviation	% papers (may belong to ≥ 1 community)	% papers classified to single community
1	Artificial Intelligence	AI	12.74	10.40
2	Algorithm and Theory	ALGO	25.47	31.34
3	Hardware and Architecture	ARC	11.9	2.92
4	Bioinformatics & Computational Biology	BIO	15.84	0.55
5	Computer Vision	CV	19.92	0.98
6	Databases	DB	7.04	14.25
7	Distributed and Parallel Computing	DIST	6.07	18.38
8	Data Mining	DM	9.82	2.89
9	Graphics	GR	0.3	1.28
10	Human-Computer Interaction	HCI	0.4	2.04
11	Information Retrieval	IR	4.1	7.32
12	Machine Learning and Pattern Recognition	ML	3.41	2.50
13	Multimedia	MUL	3.39	0.95
14	Natural Language and Speech	NLP	3.06	2.53
15	Networking	NW	10.07	7.80
16	Operating Systems	OS	23.64	3.81
17	Programming Languages	PL	4.57	7.53
18	Real Time Embedded Systems	RT	11.03	4.76
19	Scientific Computing	SC	7.05	0.43
20	Software Engineering	SE	9.84	2.40
21	Security and Privacy	SEC	9.38	1.20
22	Simulation	SIM	13.82	4.08
23	World Wide Web	WWW	8.36	4.66

1. *Preprocessing of citation contexts.*
 - 1: **for** each paper P **do**
 - 2: **for** each context s in S **do**
 - 3: $s1 \leftarrow$ Remove stop words form s
 - 4: $s2 \leftarrow$ Stem all words in $s1$ using Porter Stemmer
 - 5: $s3 \leftarrow$ Generate $n -$ grams upto 3 using $s2$
 - 6: **end for**
 - 7: **end for**
2. *Count the number of times a particular n -gram occurs for a community.*
 - 1: **for** each n -gram n_j possible **do**
 - 2: Initialize 23 counters, one for each research community, to 0.
 - 3: **end for**
 - 4: **for** each community C_i a paper P belongs to **do**
 - 5: **for** each n -gram n_j in $s3$ for paper P **do**
 - 6: Increment count of n_j for community C_i
 - 7: **end for**
 - 8: **end for**
3. *Calculate idf as follows. ($idf_{i,j}$ denotes idf of n -gram n_i for community C_j)*
 - 1: **for** each n -gram n_j possible **do**
 - 2: **for** each of the 23 communities C_j **do**
 - 3: $idf_{i,j} \leftarrow \log$

$$\frac{\text{count total papers in } C_j \text{ for the data set}}{\text{count of occurrence of } n\text{-gram } n_i \text{ for community } C_j \text{ (as calculated in step 2)}}$$
 - 4:
 - 5: **end for**
 - 6: **end for**
4. *Classify each paper into a single community.*
 - 1: **for** each paper P **do**
 - 2: Initialize counters to keep score for all 23 communities to 0
 - 3: **end for**
 - 4: **for** each n -gram n_i in $s3$ for paper P **do**
 - 5: **for** each of the 23 communities C_j **do**
 - 6: Increase community score for C_j for paper P by $idf_{i,j}$
 - 7: **end for**
 - 8: **end for**
 - 9: Paper P is classified in to community which has the maximum score

3 Construction of the Citation Network

3.1 Preprocessing

The citation context were filtered to remove stop-words. Keywords from all the articles were consolidated to form a global dictionary of about 400000 terms.

3.2 Citation Network

A citation network is formed with articles as nodes, and an edge from article i to article j iff i contains a citation to j , where $i, j \in A$. We form a direct citation network from the dataset using ‘citations.txt’ where we have one DOI citing another DOI. The Citation network we created consists of 407542 edges.

3.3 Term Labelled Citation Network

Let A be the set of all articles. Let T be the global set of all terms used in all articles in A . A term-labelled citation network Fig. 3, denoted by $G(A, C)$, is a directed graph with set of edges $C = A \times A$ where $(i, j) \in C$ iff article ‘ i ’ cites article ‘ j ’. The edge (i, j) is labelled with all terms in T_{ij} where $T_{ij} \subset T$ and T_{ij} is the set of all terms that appear in at least one citation context in article i to article j . Note also that $T_{ij} = \phi$, if there is no citation from article i to article j , or the citation context has no term in it.

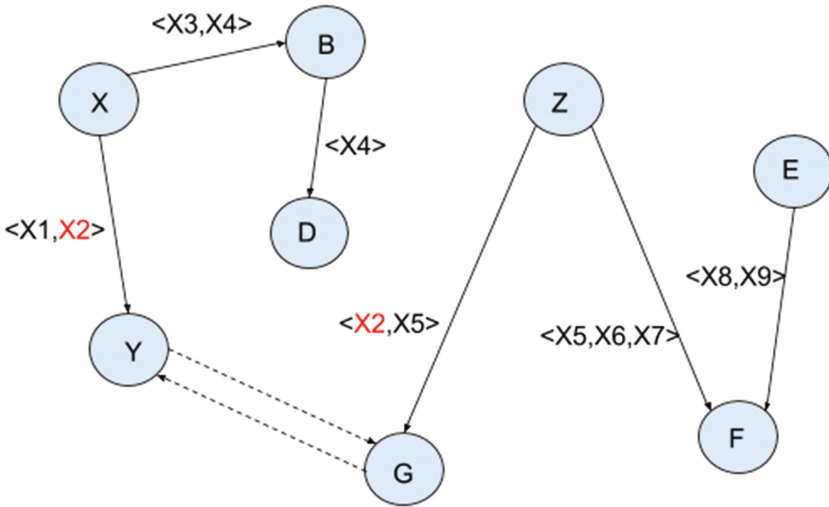


Fig. 3. Term-labelled citation network

4 Proposed Methodology

4.1 Tagging Papers with a Set of Keywords

I We merge the citation contexts for each paper. Suppose citations data contains a row $(DOI_from, DOI_to, citationContext) \Rightarrow (i, j, C_k)$ then for each paper j , we merge C_k ’s of all rows where j appears in DOI_to column.

- II We extract keywords from each filtered citation context using global set, T . For each unique paper considered in our study, we take the union of all such extracted terms. Using this data, we create a Term-Labeled Citation Network for our dataset.
- III In the Term-Labelled Citation Network, we pick up each paper 'j' $\in A$, and for every $(i, j) \in C$, we form a union U_j of the terms in set T_{ij} .
Now, U_j contains a set of all terms that describe 'j' i.e. they exist at least once in any citation context of all papers 'i' that cite paper 'j'.
- IV For each unique DOI 'j' $\in A$, we combine U_j with the already existing set of keywords for that paper.

4.2 Similarity of Term Sets

Let G_i denote the articles that are directly cited by article 'i' and T_i be the set of terms defining the article 'i'.

Once we have all terms defining each article 'i' denoted by T_i , a term set comparison is done of every article i with each article j, where $j \in (A - G_i)$. We determine the similarity between term set T_i and each term set T_j and rank articles in $(A - G_i)$ in decreasing order of similarity. Keeping a minimum threshold of n common terms, we extract the top five (or less) articles in the ranking and make edges (both inbound and outbound) between i and these articles as can be seen from the dotted lines between the papers 'G' and 'Y' in Fig. 3.

The improvised citation network after considering citation context analysis gives about 6500 additional edges, which were earlier not part of the citations dataset.

5 Community Network

A community network is a directed weighted graph and can be defined formally as a graph $G(V, E)$ where,

V = set of vertex, each vertex representing a community

E = set of edges, there exists an edge from community u to v if a paper in community u cites some paper in community v

w_E = edge count, an edge from u to v has weight w if is the total number of citations from any paper in u to any paper in v.

A community network is created on top of the citation network with the aid of the fields annotated to the papers. This is further used to represent the interaction between the research communities, to show how one community depends on other communities for growth or may be helps them to grow instead.

We create community graphs on traditional as well as context based citation networks, then perform metrics evaluation on both and compare the results to validate improvements on the traditional community network proposed in [2].

6 Metrics Evaluation and Comparison

The scoring functions [2] that have been evaluated to characterize how “community-like” is the connectivity structure of nodes in a community network are **Expansion (EXPN)**, **Cut Ratio (CUT)**, **Fraction over median degree (FOMD)**, **Conductance (COND)**, **Flake-ODF (ODF)** and **Inwardness (INWD)**.

6.1 Detailed Analysis of the Transitions in Graph

The proposed algorithm has been run using a minimum threshold value of 5 and then 7 for the similarity index of two papers, i.e. when the $|\text{keywords of paper 1} \cap \text{keywords of paper 2}| > \text{threshold value}^2$. For example consider 2 non adjacent

Table 2. Percentage change in metric values of the improvised network with respect to that of the citation network (using threshold 5)

	FOMD	COND	CUT	EXPN	INWD	ODF
AI	23.91111111	0.05140325205	7.060654101	7.060654101	6.249464699	24.64040025
ALGO	23.84982639	-0.1863917373	8.078769207	8.078769207	6.639635477	26.59033079
ARC	28.61271676	-0.1679844694	5.048629002	5.048629002	5.083101063	32.13773314
BIO	33.33333333	0.02677645011	2.617397998	2.617397998	3.986710963	61.11111111
CV	21.88841202	-0.03512541487	3.779665983	3.779665983	2.638629536	30.96774194
DB	28.0620155	-0.2875886608	6.237638428	6.237638428	7.490801778	26.13156307
DIST	26.7817194	-0.1678107866	4.926427214	4.926427214	5.500712353	25.48828125
DM	24.68856172	-0.1342349124	6.890195972	6.890195972	6.836566182	26.5323993
GRP	22.42314647	0.05707913097	3.833107191	3.833107191	3.514326421	33.5243553
HCI	29.14171657	0.02311113077	3.107505456	3.107505456	2.923371449	35.88235294
IR	27.6340694	-0.087180401	6.222726837	6.222726837	6.358722069	35.72093023
ML	21.56156156	-0.0518662961	4.746119338	4.746119338	5.245283019	28.994614
MUL	26.9047619	-0.0872297783	3.013976315	3.013976315	3.142991423	35
NLP	24.58857696	0.547107432	7.473979845	7.473979845	4.954217126	31.62274619
NW	26.91867125	-0.4421937608	4.938986372	4.938986372	5.815988499	32.03342618
OS	23.41434499	-0.04005713242	4.884920581	4.884920581	4.372213247	23.41376229
PL	24.63942308	-0.09315561805	7.641857792	7.641857792	6.460235671	27.01271186
RT	23.19376026	-0.03205419793	5.277127752	5.277127752	4.589043873	25.70332481
SC	21.27659574	-0.003138646069	4.688332445	4.688332445	5.389755011	30.76923077
SE	28.70813397	-0.1995223817	3.857820833	3.857820833	5.116126532	35.32818533
SEC	28.82882883	-0.08211246457	3.391346	3.391346	4.915055432	29.61275626
SIM	23.42901474	0.01965921689	3.846070555	3.846070555	3.307605093	31.63972286
WWW	28.16386247	-0.08903958947	4.242135367	4.242135367	6.499169783	36.11416026

² keywords here represent the set of keywords mentioned in papers + keywords extracted from the citation contexts of the paper.

papers P1 with keywords {X1, X2, X4, X5, X7, X9, X11} and P2 with keywords {X1, X2, X5, X7, X9, X10, X11}. If the threshold is taken to be 5, an edge will be added from P1 to P2 (and also from P2 to P1) in the improvised citation network because the intersection of their term sets results in {X1, X2, X5, X7, X9, X11} which has cardinality $6 > 5$. Table 2 shows the results of our study (considered interdisciplinary behaviour of research papers) in terms of percentage change in the metric values based on which the following inferences were drawn:

1. Cs and Ms values have increased for all communities due to the addition of new edges in the network. Expansion and inwardness values have increased in the improved network, depicting that even at the node level the inter-community edges have increased supporting the fact that there is a slightly higher influence of papers from other communities for the growth of a particular community and vice-versa as compared to the study previously done.
2. The higher inwardness values for all the communities itself brings out the fact that the degree of authoritativeness of communities like DB, AI, ALGO, IR and WWW have shown a rise in figures because of increased number of citations received from papers from other communities. We can therefore infer that the papers have shown an increased interdisciplinary study behaviour.
3. The conductance values have increased for six communities i.e. AI, BIO, GRP, HCI, NLP and SIM and decreased for all others indicating the fact these six communities are growing by citing more number of papers from outside their own community and are less involved in research work within themselves. This can be a major indication of the possibilities of collaborative research.
4. A high FOMD value suggests that a community's in-citations from nodes within the community is high showing that there are more chances of presence of potential candidates to being seminal papers in the community which may have been overlooked in the earlier analysis. The FOMD values have increased for all the twenty-three communities.
5. The values of cut-ratio have also shown an increase for all communities thereby implying an even more increase in the number of inter-community edges in the context based citation network.
6. The Flake-ODF values have also gone up for all the communities depicting the fact that there are now more papers in a community that are citing papers from other communities more than papers from their own community.

Considering 7 as threshold value, not much difference was observed which clearly shows that above a certain value even if there is an increase in threshold value, very few new edges are added to the citation network and the study is not much affected. This leads to a conclusion that there will come a certain point when increasing the threshold will bring back the original citation network again.

The metric values when single community classification is done and threshold is kept as 5 is shown in Table 3 (Figs. 4, 5, 6 and 7).

Table 3. Percentage change in metric values of the improvised network with respect to that of the citation network (using threshold 5 and single community classification)

	FOMD	COND	CUT	EXPN	INWD	ODF
AI	6.0322854715	-0.0027325566	4.404185538	4.404185538	5.762283237	33.0860534125
ALGO	6.5474808451	-0.0654558965	6.9072124604	6.9072124604	5.4276315789	27.6951672862
ARC	9.8214285714	0.3847596709	2.6654619381	2.6654619381	5.2331432405	50
BIO	4.7619047619	-0.2610038399	2.7586206897	2.7586206897	4.8850574713	52.9411764706
CV	5.2083333333	1.3069814398	2.3952095808	2.3952095808	2.135678392	52.7777777778
DB	9.0573012939	0.3118670996	5.1963210234	5.1963210234	7.0044361429	35.2760736196
DIST	7.8204534938	0.2118237223	5.1432738504	5.1432738504	4.9120492524	30.7913669065
DM	6.6844919786	-0.0732215468	4.1675571703	4.1675571703	6.1633663366	22.8187919463
GRP	5.5555555556	0.3316426063	3.6393713813	3.6393713813	4.6181172291	25.4901960784
HCI	6.9565217391	0.3385920271	2.3359840954	2.3359840954	5.3333333333	106.4516129032
IR	8.0254777707	0.2155716088	6.3340304751	6.3340304751	5.8067327596	46.9072164948
ML	7.2727272727	0.1194025539	3.1431187061	3.1431187061	5.6475903614	35.3982300885
MUL	15.5555555556	0.201857878	1.4692378329	1.4692378329	5.0243111831	100
NLP	3.7542662116	-0.0806210809	3.4090909091	3.4090909091	3.3772652389	52.5614754098
NW	13.4099616858	0.9158179949	3.2544080605	3.2544080605	10.4537938637	22.9885057471
OS	5.2854122622	0.08470524	3.5650040883	3.5650040883	3.4665782053	64.2857142857
PL	5.6131260794	0.0832700957	8.8361164723	8.8361164723	5.3833719868	34.7058823529
RT	6.1692969871	0.0357255429	6.4995477452	6.4995477452	3.8733171485	26.368159204
SC	8.5106382979	0.4667832168	4.8442906574	4.8442906574	1.821192053	28.2352941176
SE	7.6923076923	0.2083569666	2.9874718921	2.9874718921	5.0420168067	30.7692307692
SEC	15.2173913043	0.7176755035	4.2228739003	4.2228739003	14.6694214876	51.4705882353
SIM	6.5708418891	-0.0475189936	4.5358038989	4.5358038989	3.9578324394	76.1904761905
WWW	6.25	-0.1424653265	4.1111534069	4.1111534069	5.3492063492	31.3953488372

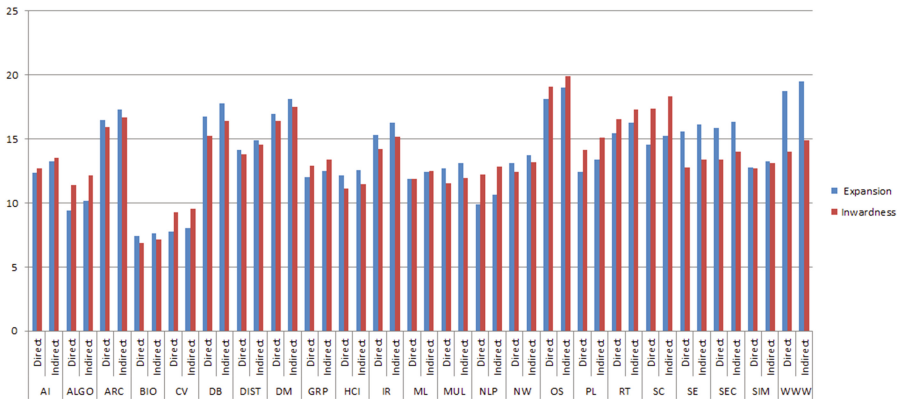


Fig. 4. Expansion and inwardness values

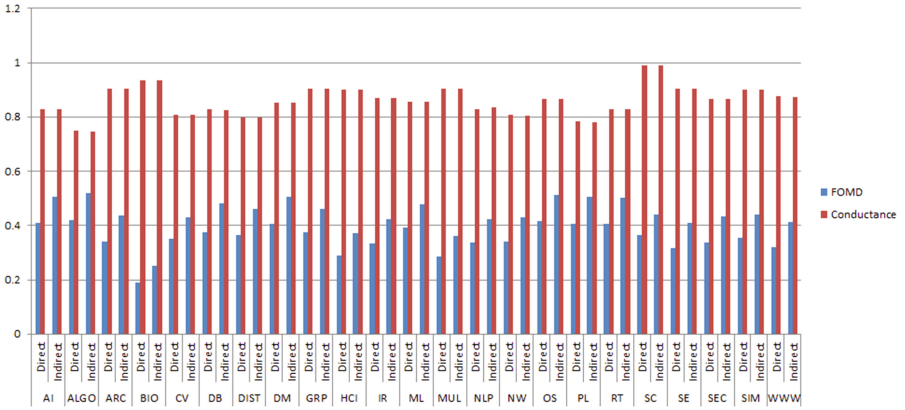


Fig. 5. FOMD and conductance values

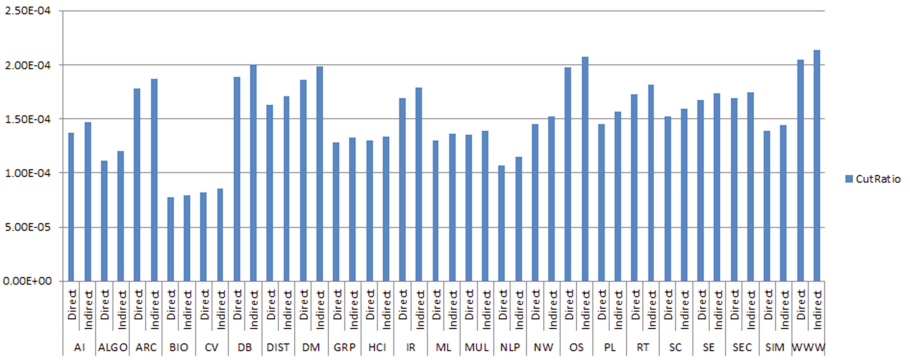


Fig. 6. Cut-ratio values

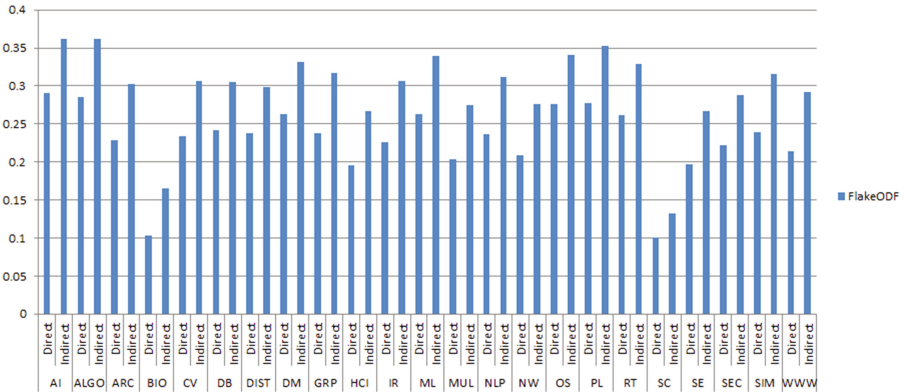


Fig. 7. Flake-ODF values

7 Conclusion

An edge in a citation network carries more information than just a single binary relation. Terms used in citation contexts can be exploited to describe the cited article in a better way without considering its actual content. Considering only direct citations constricts us from taking into account all those articles which were similar to the citee but were not directly cited and hence include them in the topic under consideration. This can make the study of citation networks more accurate and brings out better results. The impact of the ground truth communities on each other could be analyzed and studied in a better manner on this improved citation network.

Several network parameters other than citation context can be taken into consideration to further enhance the study. A network formed as a result can improve the study to a larger extent and at the same time bring out some more patterns in the impact of these ground truth communities over each other. Our study had some limitations (like small dataset) which can be improved to better emphasize the facts that we could show only on a small scale. As of now the study is limited to computer science sub-communities, but with a much bigger dataset it can be expanded to highlight characteristics and limiting factors of citation and community networks for broader fields of study. Temporal analysis can be done to understand the change in the metric values over time for all the communities and study the impact, rise and fall of the communities with time. In addition to using ground truth to form communities, we could also use verification mechanism to ensure that the newly formed communities are more authentic and sensible. Outlier analysis on the networks might also bring out interesting results.

References

1. Chen, P., Redner, S.: Community structure of the physical review citation network. *J. Inf.* 4(3), 278–290 (2010). <http://www.sciencedirect.com/science/article/pii/S1751157710000027>
2. Chakrabort, T., Sikdar, S., Tammana, V., Ganguly, N., Mukherjee, A.: Computer science fields as ground-truth communities: their impact, rise and fall. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2013, New York, NY, USA, pp. 426–433. ACM (2013). <http://doi.acm.org/10.1145/2492517.2492536>
3. He, Q., Pei, J., Kifer, D., Mitra, P., Giles, L.: Context-aware citation recommendation. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, New York, NY, USA, pp. 421–430. ACM, (2010). <http://doi.acm.org/10.1145/1772690.1772734>
4. Huang, W., Wu, Z., Chen, L., Mitra, P., Giles, C.L.: A neural probabilistic model for context based citation recommendation. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 25–30 January 2015, Austin, Texas, USA, pp. 2404–2410 (2015). <http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9737>

5. Yang, J., Leskovec, J.: Defining and evaluating network communities based on ground-truth. *Knowl. Inf. Syst.* **42**(1), 181–213 (2015). New York, NY, USA: Springer-Verlag. <https://doi.org/10.1007/s10115-013-0693-z>
6. Caragea, C., Bulgarov, F.A., Godea, A., Gollapalli, S.D.: Citation-enhanced keyphrase extraction from research papers: a supervised approach. In: *EMNLP*, vol. 14, pp. 1435–1446 (2014)

Mining Informative Words from the Tweets for Detecting the Resources During Disaster

Madichetty Sreenivasulu^(✉) and M. Sridevi

Department of Computer Science and Engineering,
National Institute of Technology, Tiruchirappalli 620015, Tamil Nadu, India
{406116004,msridevi}@nitt.edu

Abstract. Millions of tweets are posted on twitter during disaster. Many prior studies discussed about detection of situational and non-situational information occurred during disaster. It is difficult task to detect the tweets related to the resources because tweets related to the resources is a subset of the situational information. During disaster the data are unlabeled. It is not possible to predict the data in supervised classifier without labels. Hence, a classifier based on the informative words for detecting the resources is proposed in this work. It is trained with past data and tested with future events. In this work, the Italy earthquake 2016 data-set is used which is provided by SMERP 2017. First day tweets are used for training the classifier and second and third day tweets are used for testing purpose. The proposed features outperforms than Bag-Of-Words (BOW) in both in-domain and cross-domain schemes.

Keywords: Resources · Twitter · Cross-domain

1 Introduction

Social media plays a vital role in communicating and to understand the situation of the events during disaster. The best examples for social media are Twitter, Facebook etc. Among them, Twitter become more popular because of real-time nature and limit of the tweet length. Most of the recent research studies [1–7] have shown the importance of twitter to understand the situational awareness during disaster. A colossal amount of information's were posted on the Twitter during disaster. Many prior works focused on detecting the situational information on the twitter during disaster [4, 5, 8–11]. Situational information includes dead or injured people, missing, trapped, or found people, displaced people, infrastructure and utilities damage, donation needs or offers or volunteering services, shelter and supplies etc., and other related information to the disaster [10]. Detecting the tweets related to the resources is a difficult task because it is a subset of the situational information [11]. Many organizations look for the information related to the resources because the limited resources cannot be provided to all the people [3]. Detecting the information related to the resources is very

helpful for the organizations to provide the needy item for the people and also victims for requesting the resources.

The detection problems are considered as an classification problem. The supervised classifiers are adopted for the problem in [6, 12, 13]. In [12], the authors developed a classifier for detecting the tweets related to earthquake during disaster for Japanese language and developing an application for reporting Japanese tweets. In [6], authors used a Convolutional Neural Network for detecting tweets related to earthquake during disaster. The authors in [13], provided 19 different crises data-sets between the years 2013 and 2015 for developing the automatic method to detect different class labels during disaster. Additionally, it provides the largest word2vec embeddings and human-annotated lexicons for different lexical variations. They are very much useful for detecting the situational information during disaster.

In this work, informative words from the twitter are mined for detecting of resources during disaster. Based on the informative words, deployed a classifier for training and testing data-set. The process is to extract the tweets related to disaster and then classify them related to the resources. After the information's are classified based on the resources, organizations can plan accordingly based on the information and victims can be helped. The main contributions of the proposed work are as follows:

1. Mining informative words from the tweets during disaster.
2. Compared the proposed work with existing Bag-Of-Words (BOW) model in both cross-domain and in-domain schemes.
3. Detecting the availability and requirement of resources during disaster with the help of informative words.

The rest of the paper is organized as follows. The related works are discussed in Sect. 2. Section 3 describes about the proposed work. Experiment results and analysis are described in Sect. 4. Finally, the paper is concluded in Sect. 5.

2 Related Work

People pose messages on twitter during disaster which are helpful to the humanitarian organizations for disaster response efforts. Many approaches are developed for detecting the tweets related to the disaster and they are explained in this section. In [4], authors developed a system for extracting the situational information from the Twitter during various disasters and crises with the use of natural language processing and data mining techniques. The features such as Uni-gram, Bi-gram, length of words, presence of re-tweets (forwarded tweet) and replied tweets (the tweet is replied from others) are considered for it. After detecting the situational information, there is a need for a summary related situational information which unit help to get situational awareness because it is difficult to read all the situational information from the tweets. Therefore, the authors in [14], developed a model by ranking the tweets in a unified mutual reinforcement graph. And also social influence of users and content quality of tweets are

also taken into consideration for ranking the tweets. The tweet summaries are produced at the end. In [10], the authors developed a framework for getting a situational awareness with the help of abstractive summaries. For generating the summaries of a tweets, there is a necessary to find the important set of tweets from the whole tweets. The important set of tweets are found from the whole information tweets by using the integer-linear programming based on optimization technique. From it, the final summary is produced with the use of word graph and concept event based abstractive summarization Technique.

During disaster, tweets includes not only situational information but also non-situational information. Communal tweets is a subset of the non-situational information. It has a great exposure during disaster when compared to non-communal tweets. The authors in [15], characterize the communal tweets by considering five recent disaster data-sets namely NEQuake, KFlood, GShoot, PAttack and CShoot. They found that the communal tweets are not only posted by the common people but also posted by many popular persons who have thousands of followers. It is found that the users who posted communal posts form a strong connected group for producing adverse effect during disaster in a social network. However, most of the prior works discussed in the literature's were focused only on situational and non-situational information's and also they worked only on in-domain schemes.

Due to lack of communication between the victims and local government or humanization organizations, there is a lot of resources were wasted during disaster. And also humanitarian organizations couldn't provide aid services to the victims. Many of the requests and necessary aid activities were reported through the twitter. Therefore, the authors in [2] developed a method for identifying the tweets related to the resources. And also use the information retrieval methods for finding the best match between the resource requirement and availability. The fair result is shown in the classification of tweets related to the resource request and availability, donation related and resource related messages. However, they used more number of features which takes more time for processing and also the method is applicable only for in-domain scheme. A method was developed in [1] for matching the tweets between the resource requests or problem reports and necessary aid activities for solving the problems during disaster. The proposed method can be applicable only to the Japanese tweets. It cannot be used for English tweets. And also the method doesn't explains about the identification of tweets related to the resources. The case study of the Nepal earthquake 2015 is described in [16] based on the whats-app chat data-set. It analyzes the types of resources needed after-effect of the earthquake. But, they doesn't provide automatic detection of the resources from the tweets during disaster.

Hence, this paper focuses on detection of resources during disaster in both in-domain and cross-domain with limited number of features. A classifier is utilized based on the informative words on a tweets to detect them.

3 Proposed Method

During 2016 Italy earthquake, 72,200 tweets were posted on Twitter. Organizations post the tweets on a twitter during disaster using the content words such as help, army, aid, earthquake, mobilized etc. for helping the people by providing the resources. People posts the tweets for the requirement of resources during disaster by using the content words like blood, victims etc. This section describes about the proposed features such as terms related to disaster, infrastructure damage, communication, location, injury and human for detecting the resources based on the tweets during disaster. Resources related to the tweets are categorized into two types. 1. Availability of resources and 2. Requirement of resources. Sections 3.1.1 and 3.1.2 explain about the tweets related and non-related to the availability and requirement of resources respectively. There is no model available for detecting the tweets related to resources based on the proposed features.

The steps involved in the proposed work are listed below:

1. Tweet collection
2. Pre-processing
3. Feature extraction & training phase
4. Testing phase

Each and every steps is explained in the Sects. 3.1 to 3.4.

3.1 Tweet Collection

Tweets corresponding to tweet id's are crawling using twitter API. The examples of tweets for related and non-related to the resources are presented in Table 1.

3.1.1 Availability and Requirement of Resources

From the collected tweets, relevant tweets are identified which mention the availability and requirement of some resources like food, drinking water, shelter, clothes, blankets, blood, volunteers (human resources), tents, water filter, power supply, etc. Tweets stating the availability of transport vehicles for assisting the resource distribution process is also considered as relevant. Tweets indicating any services like free WIFI, SMS, calling facility etc., is also relevant. In addition, any tweet or announcement related to donation of money will also be relevant.

3.1.2 Non-availability and Requirement of Resources

General statements without referring to any resource is considered as a non-relevant information. It may be include the tweets related to infrastructure damages, injured people and affected people. It also include tweets which are non-related to disaster event.

3.2 Pre-processing of Tweets

After crawling, the tweets are preprocessed for reducing the complexity of the classifier. The steps involved in pre-processing is explained below:

1. Normalization and Tokenization: Normalization is a process of changing all the letters of the tweets to lower case. Dividing the tweets into the number of tokens is called as Tokenization.
2. Stop-Words: The words are used commonly in all the tweets are called Stop-words. Those stop-words has to be removed from the tweets.
3. Removal of numerals, URL’s, hash-tags(#), user-mentions(@) and unknown symbols from the tweets has to performed.

Table 1. Some example of tweets for related and non-related to availability and requirement of resources

<p>Availability of resource related tweets in SMERP LEVEL-1</p> <p>#ItalyEarthquake we are sending food, water and medicine to survivors of the 6.2 magnitude earthquake @Redcross [URL]</p> <p>A @ShelterBox response team will be in Italy within 24 h, to assess the need for emergency shelter in Italy after today’s #earthquake</p>
<p>Availability of resource related tweets in SMERP LEVEL-2</p> <p>KFC, MC Donald join family eateries at home and abroad in donating food and funds after Italy Earthquake [URL]</p> <p>Photo: Asylum seekers volunteer to assist in Italy earthquake rescue mission @RadioRia via @breaking [URL]</p>
<p>Requirement of resource related tweets in SMERP LEVEL-1</p> <p>Rieti hospital in Italy is asking for blood donors of all blood types #terremoto #italyearthquake [URL]</p> <p>I’m raising money for Help For Earthquake in Amatrice, IT. Click to Donate: [URL] via @gofundme</p>
<p>Requirement of resource related tweets in SMERP LEVEL-2</p> <p>Donate to day to help save Victims or provide Food Water etc. [URL]</p> <p>Italians urged to Remove Wifi passwords to Help Earthquake Victims. Disaster communications [URL]</p>
<p>Non-Related resources tweets</p> <p>BREAKING: A magnitude 6.4 #earthquake has just hit #Italy near #Perugia. No words on damage or fatalities yet. More to come</p> <p>“No immediate reports of damage in quake that rattles Italy: [URL]</p> <p>@philsnews @eCapitol_Shawn @Okelections I go to Twitter for Oklahoma election and earthquake results</p> <p>What we know so far about the Italy quake, which has been revised to 6.2: [URL] [URL]</p> <p>The best good morning? Being wake up by an earthquake of course</p>

3.3 Feature Extraction and Training Phase

After pre-processing is done on the tweets, the proposed features namely disaster, infrastructure damage, communication, location, injury and human related terms are extracted from the preprocessed tweets. The proposed features are also known as informative words. The proposed features is given as an input to the Support Vector Machine (SVM) classifier to train and test the disaster data-set (Tables 2 and 3).

Table 2. Proposed features for detecting the resources based on tweets

S. No.	Features (Terms related to)	Explanation (Tweet contains)
(1)	Disaster	Terremoto, Italy earthquake, quake, earthquake, magnitude, major and hit terms
(2)	Infrastructure damage	Buildings, breaking, causalities, damage, damaged, and infrastructure damage terms
(3)	Communication	Wifi, news, passwords, reported, report, restoration and causalities terms
(4)	Location	Rome, Italy, near, cross and town terms
(5)	Injury	Blood, dead, aid, affected, donate and help terms
(6)	Human	The terms are army, italian, people, victims, injuries and rescue terms

Table 3. Proposed features with example tweets

S. No.	Features	Examples of features related to informative words
(1)	Disaster	Our condolences and sympathy to everyone affected by the #Italy earthquake. We stand ready to help with water purification units
(2)	Infrastructure damage	#Earthquake in Italy: at #Amatrice town high damages, streets blocked and one bridge fell down. Mayor ask for help [URL]
(3)	Communication	The Red Cross Wants Italians in Earthquake Disaster Area to Deactivate Wi-Fi Passwords: Enter your username a [URL]
(4)	Location	#ItalyEarthquake #RedCross asks folks near epicenter disable wifi passwords, help 1st responders communicate: @CNN [URL]
(5)	Injury	Terrible earthquake in central Italy Shocked. Our Civil Protection Department needs blood. Please help. #terremoto
(6)	Human	Red Cross requests Italians deactivate Wi-Fi router passwords to ease earthquake rescue efforts [URL]

3.4 Testing Phase

The proposed work is tested with BOW (Bag-Of-Words) model for the following schemes:-

Scheme 1: In-domain & Scheme 2: Cross-domain

In-Domain. The training and testing of the proposed work is performed with the same disaster event data-set, then it is called In-domain.

Cross-Domain. Cross-domain means performing the training and testing on different disaster event data-set. Train the model with the past event data-set and tested the model with the future event data-set. It is very much helpful when the labeled data are limited.

4 Experimental Results and Analysis

The proposed work is implemented in Python Language [17]. SVM classifier [18] with default RBF kernel is used for classifying the disaster data-set both in-domain and cross-domain schemes. The details of the data-set which is used for this problem is described in the Sect. 4.1. The proposed work is compared with the existing BOW model by considering different parameters such as accuracy, auc_roc score, precision, F1-score and training time of the classifier. The proposed work is implemented in both in-domain and cross-domain schemes and their performance is evaluated in Sect. 4.2.

4.1 Data-Set

The SMERP 2017 data-set is used for implementing the proposed work. SMERP 2017 contains tweet id's of the tweets which are posted on twitter at the time of Italy earthquake August 2016 and it is provided by the track organizers. Tweet id's are categorized into two levels. Level-1 tweet-id's represent the tweets posted during first 24 h after the earthquake. Level-2 tweet-id's represent the tweets posted during second and third day after the earthquake. Total number of tweets present in level-1 is 52,469 and level-2 contains 19,751 with 4 topics in the TREC format [19].

Table 4. Comparison of proposed features with different combination of n-gram features for SMERP level-1

Proposed features with	Precision	Recall	F1-score	Accuracy	Auc.Roc score
Uni-grams	84	96	90	90.3	90.9
Bi-grams	59	98	74	69.3	72.2
Trigrams	47	100	64	49.1	54.3
Both Uni-grams, Bi-grams	85	96	91	91.1	91.6
Uni-grams, Bi-grams & Trigrams	85	96	91	91.1	91.6

Table 5. Comparison of proposed features with the different combination of n-gram features for SMERP level-2

Proposed features with	Precision	Recall	F1-score	Accuracy	Auc_Roc score
Uni-grams	85	100	92	91.6	92.2
Bi-grams	56	95	71	63.6	65.8
Trigrams	48	97	64	50.0	53.2
Both Uni-grams & Bi-grams	85	100	92	91.6	92.2
Uni-grams, Bi-grams & Trigrams	85	100	92	91.6	92.2

4.2 Performance Measures

The proposed features are compared with different combination of n-gram features such as Uni-gram, Bi-gram and trigram for SMERP level-1. And their results are tabulated in Table 4. First column shows the proposed features with different combinations of n-gram features. First row indicates names of the parameters. The proposed features with the combination of Uni-grams and Bi-grams give better values when compared to other combination of features. Inclusion of trigram features doesn't have much impact on the result. The proposed features with the combination of Bi-gram and trigram features doesn't produce good result.

Table 6. Comparison of Proposed features (PRO) as Uni-grams with Bag-Of-Words (BOW) for Precision value

Training data-set	Testing data-set			
	SMERP level-1		SMERP level-2	
	BOW	PRO	BOW	PRO
SMERP level-1	44	84	46	88
SMERP level-2	44	80	46	85

Similarly, the comparison is made with different combination of n-gram features for SMERP level-2. In this case, combination of Uni-grams provides better result and the results are tabulated in Table 5.

Table 7. Comparison of Proposed features (PRO) as Uni-grams with Bag-Of-Words (BOW) model for F1-score parameter

Training data-set	Testing data-set			
	SMERP level-1		SMERP level-2	
	BOW	PRO	BOW	PRO
SMERP level-1	61	90	63	94
SMERP level-2	61	87	63	92

Table 8. Comparison of Proposed features (PRO) with Bag-Of-Words (BOW) model for Accuracy parameter

Training data-set	Testing data-set			
	SMERP level-1		SMERP level-2	
	BOW	PRO	BOW	PRO
SMERP level-1	44.3	90.3	46.2	93.9
SMERP level-2	44.3	87	46.2	91.6

Table 9. Comparison of Proposed features (PRO) with Bag-Of-Words (BOW) model for AUC_ROC Score

Training data-set	Testing data-set			
	SMERP level-1		SMERP level-2	
	BOW	PRO	BOW	PRO
SMERP level-1	50.0	90.9	50.0	94.3
SMERP level-2	50.0	87.8	50.0	92.2

Table 10. Comparison of Training time of the Proposed features (PRO) with Bag-Of-Words (BOW) model

Training data-set	Testing data-set			
	SMERP level-1		SMERP level-2	
	BOW	PRO	BOW	PRO
SMERP level-1	152 ms	6 ms	144 ms	6 ms
SMERP level-2	176 ms	4 ms	180 ms	6 ms

The proposed features are compared with BOW model in both in-domain and cross-domain schemes with the parameters such as precision, accuracy, F1-score and Auc_Roc score and the results are shown from Tables 6, 7, 8 and 9 respectively. First column in table represents the data-set used for training and second row in table represents the data-set used for testing. During disaster, there is a need to reduce the training time for supervised classifier. Proposed features reduces the training time compared to BOW model and it is tabulated in Table 10. Proposed features outperforms well than the existing BOW model for all parameters.

5 Conclusion

Social media often serves as a communication media during disaster. Most of tweets contains the information related to the disaster and it includes the information about the availability and requirement of the resources. During disaster,

for detecting the tweets related to the resources, there is a need of labeled data for supervised learning algorithm but getting labeled data is very difficult during disaster. A method has been proposed based on the informative words of a tweets for detecting the resources. And also training the labels with past data-set and predicting the labels with future data-set. Proposed features outperforms than the BOW model both in-domain and cross-domain schemes.

References

1. Varga, I., Sano, M., Torisawa, K., Hashimoto, C., Ohtake, K., Kawai, T., Jong-Hoon, O., De Saeger, S.: Aid is out there: looking for help from tweets during a large scale disaster. *ACL* **1**, 1619–1629 (2013)
2. Purohit, H., Castillo, C., Diaz, F., Sheth, A., Meier, P.: Emergency-relief coordination on social media: automatically matching resource requests and offers. *First Monday* **19**(1), 1–6 (2013)
3. Vieweg, S., Castillo, C., Imran, M.: Integrating social media communications into the rapid assessment of sudden onset disasters. In: Aiello, L.M., McFarland, D. (eds.) *SocInfo 2014*. LNCS, vol. 8851, pp. 444–461. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13734-6_32
4. Yin, J., Lampert, A., Cameron, M., Robinson, B., Power, R.: Using social media to enhance emergency situation awareness. *IEEE Intell. Syst.* **27**(6), 52–59 (2012)
5. Imran, M.: Extracting information nuggets from disaster-related messages in social media (2013)
6. Nguyen, D.T., Al Mannai, K.A., Joty, S., Sajjad, H., Imran, M., Mitra, P.: Rapid classification of crisis-related data on social networks using convolutional neural networks. arXiv preprint [arXiv:1608.03902](https://arxiv.org/abs/1608.03902) (2016)
7. Imran, M., Elbassuoni, S., Castillo, C., Diaz, F., Meier, P.: Practical extraction of disaster-relevant information from social media. In: *Proceedings of the 22nd International Conference on World Wide Web*, pp. 1021–1024. ACM (2013)
8. Cameron, M.A., Power, R., Robinson, B., Yin, J.: Emergency situation awareness from twitter for crisis management. In: *Proceedings of the 21st International Conference on World Wide Web*, pp. 695–698. ACM (2012)
9. Caragea, C., Silvescu, A., Tapia, A.H.: Identifying informative messages in disaster events using convolutional neural networks. In: *International Conference on Information Systems for Crisis Response and Management* (2016)
10. Rudra, K., Banerjee, S., Ganguly, N., Goyal, P., Imran, M., Mitra, P.: Summarizing situational tweets in crisis scenario. In: *Proceedings of the 27th ACM Conference on Hypertext and Social Media*, pp. 137–147. ACM (2016)
11. Rudra, K., Ghosh, S., Ganguly, N., Goyal, P., Ghosh, S.: Extracting situational information from microblogs during disaster events: a classification-summarization approach. In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 583–592. ACM (2015)
12. Sakaki, T., Okazaki, M., Matsuo, Y.: Tweet analysis for real-time event detection and earthquake reporting system development. *IEEE Trans. Knowl. Data Eng.* **25**(4), 919–931 (2013)
13. Imran, M., Mitra, P., Castillo, C.: Twitter as a lifeline: human-annotated Twitter Corpora for NLP of crisis-related messages. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, May 2016. European Language Resources Association (ELRA) (2016)

14. Duan, Y., Chen, Z., Wei, F., Zhou, M., Shum, H.-Y.: Twitter topic summarization by ranking tweets using social influence and content quality. In: Proceedings of the 24th International Conference on Computational Linguistics, pp. 763–780 (2012)
15. Rudra, K., Sharma, A., Ganguly, N., Ghosh, S.: Characterizing communal microblogs during disaster events. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 96–99. IEEE (2016)
16. Basu, M., Ghosh, S., Jana, A., Bandyopadhyay, S., Singh, R.: Resource mapping during a natural disaster: a case study on the 2015 Nepal earthquake. *Int. J. Disaster Risk Reduction* **24**, 24–31 (2017)
17. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
18. Vapnik, V.N., Vapnik, V.: *Statistical Learning Theory*, vol. 1. Wiley, New York (1998)
19. Ghosh, S., Ghosh, K., Chakraborty, T., Ganguly, D., Jones, G., Moens, M.-F.: First international workshop on exploitation of Social Media for Emergency Relief and Preparedness (SMERP). In: Jose, J.M., Hauff, C., Altingovde, I.S., Song, D., Albakour, D., Watt, S., Tait, J. (eds.) *ECIR 2017. LNCS*, pp. 779–783. Springer, Cham (2017). <https://doi.org/10.1007/978-3-319-56608-5>

An Ensemble Based Method for Predicting Emotion Intensity of Tweets

Sreekanth Madisetty^(✉) and Maunendra Sankar Desarkar

Department of Computer Science and Engineering,
Indian Institute of Technology Hyderabad, Hyderabad, India
{cs15resch11006,maunendra}@iith.ac.in

Abstract. Recently, user generated contents have increased tremendously in social media. Twitter is a popular micro-blogging platform in which users share their feelings, opinions, feedback, etc. It has been observed that microblogs are often associated with emotions. Several studies have focused on assigning a given tweet to one of the available emotion categories (e.g., *anger*, *fear*, *joy*, *sadness*). It is often useful in applications to find the intensity of emotion in the tweets. The focus on identifying emotion intensity is less in the literature. In this paper, we focus on determining the level of emotion intensity in the tweets. We use an ensemble of three methods: Convolution Neural Networks (CNN) with word embedding features, XGBoost with word n-gram and char n-gram features, and Support Vector Regression (SVR) with lexicon and word embedding features. The final prediction of the given tweet is obtained by the average of predictions of individual methods in the ensemble. The performance of ensemble is better than the methods in the ensemble due to diverse features. Our experimental results outperform baseline methods.

Keywords: Social media · Sentiment analysis · Emotion detection

1 Introduction

Nowadays social media plays a very important role among Internet users. Twitter is a social micro-blogging platform where people express their opinions, feelings, arguments about different topics across the world. Tweets often contain sentiments and emotions expressed by the users. Several lines of research that focus on tweets try to understand emotion or sentiment attached to it. Sentiment analysis describes whether the tweet is positive, negative or neutral. Emotion detection assigns the tweet to one of the given emotion categories (*anger*, *fear*, *joy*, and *sadness*). Existing research in this context have mainly focused on either sentiment analysis or emotion detection in Twitter [1, 27]. The focus on emotion intensity prediction is limited in the literature. It is often useful to find the intensity of emotion in text in various applications, e.g., crisis management, product quality, event reporting, etc.

In this paper, we focus on the following problem: Given a tweet and emotion category, predict the intensity of that emotion in the tweet. We use three different families of machine learning algorithms, Convolution Neural Networks (CNN), XGBoost, and Support Vector Regression (SVR) to find the emotion intensity in the tweets. Each algorithm is very popular in handling various machine learning tasks. The predictions of each algorithm are averaged to get the final prediction.

Recently, a dataset is published in WASSA-2017 shared task in emotion intensity [17] where the tweets are labeled with four emotion categories, *anger*, *fear*, *joy*, and *sadness*. For each tweet, the intensity of that emotion is also provided. Few example instances from that dataset are presented in Table 1. We use this dataset to evaluate our proposed method.

Table 1. Example tweets showing emotion intensity

Id	Text	Category	Intensity
10000	I asked for my parcel to be delivered to a pick up store not my address #fuming #poorcustomerservice	anger	0.896
20000	Job interview in the afternoon #nervous #ek	fear	0.917
30000	Today I reached 1000 subscribers on YT!, #goodday, #thankful	joy	0.926
40000	My #Fibromyalgia has been really bad lately which is not good for my mental state. I feel very overwhelmed #anxiety #bipolar #depression	sadness	0.946

Rest of the paper is organized as follows. Related literature for current work is described in Sect. 2. Next in Sect. 3, problem statement of our work is defined. Details of the proposed method are presented in Sect. 4. Experimental evaluation of the method is described in Sect. 5. We conclude the work by providing directions for future research in Sect. 6.

2 Related Work

A large amount of work has been done to detect sentiments from twitter data. Although, sentiment analysis is different from emotion intensity prediction, features which are used in sentiment analysis can also be used in emotion intensity prediction. Hence, in this section, we present related work from literature for both sentiment analysis and emotion intensity prediction tasks.

Sentiment Analysis: Part-of-speech tag, lexicons, bag-of-words, emoticons, linguistic features, semantic features, etc. are some of the common features used in sentiment analysis. A hybrid approach which uses both corpus-based and dictionary-based methods to find the semantic orientation of the opinion words

in tweets is described in [13]. Agarwal et al. [1] used POS-specific prior polarity features and tree kernel for sentiment analysis. Bag-of-words features along with Sentiwordnet, lexicons, emoticons, etc. are used in [25]. Semantic feature is added along with traditional features for sentiment analysis in [28]. Kouloumpis et al. [12] used linguistic features and lexical resources. However, in all the above methods emotion category is not considered.

Emotion Detection: A method with distant supervision for emotion classification is described in [26]. The public mood is modeled using Twitter messages in [4]. A dataset for emotion detection in Twitter is developed in [27]. The authors have considered seven emotion categories, namely, *anger*, *disgust*, *fear*, *joy*, *love*, *sadness*, and *surprise*. Another large dataset containing instances of $\langle \textit{tweet}, \textit{emotion category} \rangle$ annotation is created in [31]. The authors have used emotion-related hashtags which are present in the tweets for the creation of dataset. They have used unigrams, bigrams, sentiment words, and part-of-speech features for emotion detection. They have also considered seven emotion categories similar to Roberts et al. [27] but used *thankfulness* category instead of *anger*. However, in all the above methods intensity of the emotion is not considered. Word-emotion association lexicon is built using crowdsourcing in [20]. An annotation scheme is introduced for finding the emotion intensities in the blog posts in [2]. A supervised framework is developed for identifying the emotional expressions and intensities in [7]. However, the emotion intensities are categorical (high, medium, and low). An ensemble method for predicting emotion intensities is described in [14]. The authors have used two SVR methods with different features and a neural network method in the ensemble. However, word embedding features are not used.

3 Problem Definition

We now briefly define the problem addressed in this paper: *Given a tweet T and an emotion E , determine the intensity $Y_{T,E}$ of emotion E felt by the author of the tweet T . $Y_{T,E}$ is a real-valued score between 0 and 1.* Here 1 is the maximum possible score, and it means the maximum amount of emotion E felt by the speaker of the tweet T . Similarly, 0 is the minimum possible score, and it means the least amount of emotion E .

4 Methodology

We model the problem of predicting emotion intensity as a regression problem. We identified three methods, namely, Convolution Neural Networks (CNN), XGBoost, and Support Vector Regression (SVR) from three different family of algorithms for this prediction. These methods are selected due to their wide acceptability in the machine learning literature for performing various predictive analytics. These three methods are combined in an ensemble to retain the predictive power of the individual algorithms as well as to exploit the synergy between them.

Tweets often contain noise in the form of slang words, elongated words, spelling mistakes, abbreviations, @ mentions, etc. The maximum length of tweet is 140 characters long. We apply the following text preprocessing steps to get better performance of the model. URLs are removed, all words are converted to lower case, @ mentions and numbers are also removed as part of the preprocessing step. These preprocessed tweets are given to each of the individual methods in the ensemble for training and testing. We now describe these methods in detail.

4.1 Convolution Neural Networks (CNN)

Convolution Neural Networks (CNN) are popular in computer vision for various tasks, e.g., face recognition, image classification, action recognition, human pose estimation, scene labeling, etc. CNNs are also used in many Natural Language Processing (NLP) tasks, named entity recognition, part-of-speech tagging, chunking, etc. We used CNN for our problem on the similar lines of approach given in [10]. CNN architecture has five layers, namely, input layer, convolution layer, pooling layer, hidden layer, and output layer.

The input to the model is tweets. Let each tweet be comprised of sequence of words: $\{term_1, term_2, term_3, \dots, term_n\}$. Then tweet vector is represented as

$$T_v = w_1 \circ w_2 \circ w_3 \circ \dots \circ w_n \quad (1)$$

Where w_i is the word embedding vector of $term_i$, and \circ is the concatenation operator. Each $w_i \in \mathbb{R}^{1 \times d}$ is associated with their corresponding pre-trained word vectors. These word embeddings can be looked up in a vocabulary of the embedding matrix $W \in \mathbb{R}^{V \times d}$, where V is the number of words in the vocabulary. Words are mapped to indices from 1 to V , and the embedding matrix is created in such a way that at index i , the word embedding corresponding to the word associated with index i is present. Tweet matrix $T_m \in \mathbb{R}^{n \times d}$ is given as input to the model where each word is represented by word embedding $w_i \in \mathbb{R}^{1 \times d}$. Glove Twitter word embeddings are used in our method. These word embeddings are publicly available¹ [23]. Tweet lengths may vary, so necessary padding is applied to have equal lengths for all the tweet vectors. Next layer is convolution layer. Convolution feature maps are created to extract emotion features. Convolution feature is calculated as follows.

$$o_i = g(\alpha \cdot w_{i:i+h-1} + \beta) \quad (2)$$

where α is a convolution filter, $\beta \in \mathbb{R}$ is bias term, h is window size, $w_{i:i+h-1}$ is the concatenation of embeddings for the terms occurring in a window of length h , from positions i to $i+h-1$, and g is a non-linear function such as the hyperbolic tangent. This convolution filter is applied to each possible window of words in the tweet to produce a convolution feature map $c \in \mathbb{R}^{n-h+1}$. Next layer is max pooling layer. The main idea in this layer is to capture most important

¹ <https://nlp.stanford.edu/projects/glove/>.

activation. Let $o_1, o_2, o_3, \dots \in \mathbb{R}$ denote the output values for our filter. Max-over-time pooling is computed as follows.

$$c = \max_i(o_i) \in \mathbb{R} \tag{3}$$

The output of max-pooling layer is given as input to the dense hidden layer. The output of hidden layer is passed through the final output layer using sigmoid as the activation function. Values output by this sigmoid activation function is emitted as the prediction of the emotion intensity for the input tweet. To avoid overfitting, dropout parameter is used.

The dataset used in our experiments contains four emotion categories. Four CNNs are used for these four emotion categories. Each CNN is trained separately for each emotion category, and emotion intensities for that category are predicted. Same configuration (filter length, number of filters, word embedding dimension size, dropout rate, number of neurons in hidden layer, number of layers, etc.) is used for all the categories to train the model. This CNN model is static where the word embeddings are not changed throughout the model.

4.2 Extreme Gradient Boosting (XGBoost)

This is the second method in the ensemble. XGBoost is based on original Gradient Boosting Machine (GBM) framework [6]. It is a supervised learning algorithm. It is a tree ensemble model and is a set of Classification and Regression Trees (CARTs). Normally, a single tree is not strong enough for classification in practice. In tree ensemble, predictions of multiple trees are added to get the final prediction. Mathematically, model is written as

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k \in F \tag{4}$$

where K is the number of trees, f is a function in the functional-space F , F is the set of all possible CARTs, x_i is training data, and \hat{y}_i is the prediction. If y_i is target variable then the objective function can be written as

$$Obj = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \tag{5}$$

The first part in the above equation is training loss and second part is regularization. Additive training is used for training the model. XGBoost is often used in many of the data science competitions. It does computations parallelly and is very fast. Word n-gram and character n-gram features are used in this model.

4.3 Support Vector Regression (SVR)

This is the third method used in our ensemble which is taken from [16]. Features used are word n-grams, char n-grams, word embeddings, and lexicons. Word

embeddings are trained from Edinburgh Twitter corpus [24] using *Word2Vec* with 400 dimensions. Lexicons used in this method are AFINN [22], BingLiu [9], MPQA [32], NRC Affect Intensity Lexicon [15], NRCWord-Emotion Association Lexicon [20], NRC10 Expanded [5], NRC Hashtag Emotion Association Lexicon [18], NRC Hashtag Sentiment Lexicon [19], Sentiment140 [19], SentiWordNet [3], and SentiStrength [29]. If the lexicon consists of categorical labels for the words then number of words matching each category in the tweet are counted. If the lexicon provides scores for the words then the scores of each word in the tweet are added. SVM Regression model is trained by using these features for each category separately and emotion intensities are predicted.

4.4 Ensemble

Ensemble methods have been proved to be very successful for classification problems. A system named Webis has achieved the best performance in SemEval-2015 subtask B, “Sentiment Analysis in Twitter” [8]. In the Netflix competition [30] and KDD Cup 2009 [21], the winners have used ensemble-based methods. There are several ways to combine the classifiers, e.g., bagging, boosting, simple averaging, majority voting, stacking, etc. We tested our methods with some of them, and simple averaging performed better than the other ensemble methods. Our ensemble method works as follows. CNN with word embedding features is trained on each category separately in the training data, and it is applied to the testing data and predictions are noted. Similarly, XGBoost with word n-gram and char n-gram features is trained, and predictions of testing data are saved. In a similar fashion, SVR with lexicon and word embedding features is trained and is applied on testing data and predictions are noted. Finally, for each tweet, the average of prediction values of individual methods is considered as final prediction.

5 Experiments

5.1 Data

The dataset used in our experiments is obtained from [16]. Statistics of the data is described in Table 2. Each row of the dataset contains id, text, emotion category, and emotion intensity as described in Table 1. The emotion intensity is a real value between 0 and 1. There are four categories of emotions, namely, *anger*, *fear*, *joy*, and *sadness*. The dataset is created by using a technique called best-worst-scaling (BWS) which improves the annotation consistency and reliable emotion intensity values.

5.2 Evaluation Metrics

In this section we describe the evaluation metrics used in our approach.

Table 2. Number of tweets in each category

Emotion	Training	Validation	Testing	All
anger	857	84	760	1701
fear	1147	110	995	2252
joy	823	74	714	1611
sadness	786	74	673	1533
All	3613	342	3142	7097

– **Pearson correlation (PC):**

It measures the correlation between two variables. Pearson correlation is calculated between predicted values and gold values. Pearson correlation coefficient is calculated as

$$PC = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (6)$$

In our problem, n is the number of test tweets, x_i is predicted emotion intensity value for i^{th} test tweet, y_i is ground truth value, \bar{x} is mean of x , and \bar{y} is mean of y .

– **Spearman rank correlation (SC):**

It measures the relationship between two rankings. Let X denote the set of actual intensity values and Y denote the set of predicted intensity values. Let X and Y are converted to ranks rgX and rgY respectively. Spearman rank correlation coefficient is calculated as

$$SC = \frac{cov(rgX, rgY)}{\sigma_{rgX} \sigma_{rgY}} \quad (7)$$

where $cov(rgX, rgY)$ is the co-variance of rank variables, σ_{rgX} , σ_{rgY} are the standard deviations of the rank variables.

Sometimes, the tweets which are having high emotion content are relevant. So, it is useful to identify the high emotion related content. To test this kind of tweets, we use two additional metrics, Pearson 0.5 to 1.0 (*PCH*) and Spearman 0.5 to 1.0 (*SCH*). Pearson 0.5 to 1.0 is calculated by considering the instances only with ground truth emotion intensities which are greater than or equal to 0.5, and the rest are ignored. Similarly, Spearman 0.5 to 1.0 is calculated.

5.3 Results and Discussions

The first method used in the ensemble is CNN. Glove Twitter word embeddings are used with dimensions, 25, 50, 100, and 200 [23]. We have used 100 as maximum sentence length, window size 3, 250 filters, hidden layer with 200 neurons, dropout 0.2 as regularization parameter in our setting. The results of CNN with 25D, 50D, 100D, and 200D word embeddings are reported in Tables 3, 4, 5, and 6 respectively. We observe that the increase in dimensions results in increase in

Table 3. CNN with Glove 25D.

Emotion	PC	SC	PCH	SCH
anger	0.540	0.511	0.410	0.386
fear	0.615	0.593	0.476	0.451
joy	0.525	0.520	0.377	0.387
sadness	0.600	0.586	0.465	0.450
Average	0.570	0.552	0.432	0.419

Table 4. CNN with Glove 50D.

Emotion	PC	SC	PCH	SCH
anger	0.616	0.581	0.493	0.477
fear	0.664	0.642	0.512	0.480
joy	0.591	0.590	0.421	0.447
sadness	0.689	0.682	0.523	0.503
Average	0.640	0.624	0.488	0.477

Table 5. CNN with Glove 100D.

Emotion	PC	SC	PCH	SCH
anger	0.672	0.644	0.537	0.518
fear	0.684	0.657	0.561	0.518
joy	0.604	0.600	0.395	0.401
sadness	0.707	0.703	0.524	0.524
Average	0.667	0.651	0.504	0.491

Table 6. CNN with Glove 200D.

Emotion	PC	SC	PCH	SCH
anger	0.670	0.639	0.548	0.540
fear	0.691	0.664	0.579	0.525
joy	0.643	0.635	0.434	0.423
sadness	0.727	0.728	0.545	0.544
Average	0.683	0.667	0.526	0.508

Table 7. XGBoost.

Emotion	PC	SC	PCH	SCH
anger	0.571	0.521	0.486	0.446
fear	0.599	0.546	0.517	0.448
joy	0.572	0.567	0.394	0.379
sadness	0.666	0.662	0.471	0.449
Average	0.602	0.574	0.467	0.431

Table 8. SVR.

Emotion	PC	SC	PCH	SCH
anger	0.636	0.627	0.502	0.472
fear	0.633	0.621	0.484	0.441
joy	0.650	0.654	0.379	0.365
sadness	0.713	0.714	0.555	0.534
Average	0.658	0.654	0.480	0.453

performance. For example, CNN with 50D performance is better than CNN with 25D. Similarly, CNN with 100D is performing better than CNN with 50D, and the performance of CNN with 200D is greater than CNN with 100D. Therefore, CNN with 200D is used in our method.

The second method used in the ensemble is XGBoost. The parameters in this method are learning rate = 0.1, number of estimators = 100, booster is gradient boosting tree, and maximum depth is 3. The results of four emotion categories are reported in Table 7. The Pearson coefficient and Spearman coefficient values are higher than CNN with 25D but lesser than the CNN with other dimensional word embeddings (50D, 100D, 200D). The final method used in the ensemble is SVR (Table 8). The parameters used in this model are linear kernel, $C = 0.001$

Table 9. Ensemble (proposed method).

Emotion	PC	SC	PCH	SCH
anger	0.718	0.696	0.609	0.584
fear	0.729	0.709	0.606	0.546
joy	0.717	0.721	0.480	0.470
sadness	0.771	0.772	0.600	0.585
Average	0.734	0.725	0.574	0.546

Table 10. Comparison of our proposed method with other approaches.

Method	PC	SC	PCH	SCH
SVR (word n-grams)	0.501	0.492	0.390	0.382
SVR (Saif et al. [16])	0.658	0.654	0.480	0.453
IMS [11]	0.722	0.712	0.514	0.503
XGBoost+CNN	0.703	0.691	0.551	0.527
XGBoost+SVR	0.704	0.697	0.540	0.513
CNN+SVR	0.724	0.713	0.558	0.535
Ensemble	0.734	0.725	0.574	0.546

(penalty term). Radial Basis Function (RBF) and polynomial kernels are also tested. However, linear kernel is performing better. The evaluation metric values are better than XGBoost and CNN with 25D and 50D. An ensemble is created by averaging the predictions of three methods described in Sect. 4 and results are reported in Table 9. It can be observed that the Pearson coefficient for both 0 to 1 and 0.5 to 1.0, the ensemble values are higher than all other methods. Similarly, the Spearman coefficient for both 0 to 1 and 0.5 to 1.0, the ensemble method values are higher.

Comparison with SVR [16], IMS [11] and combination of methods in the ensemble is presented in Table 10. We observe that our ensemble method significantly outperforms the baselines and other combinations. This shows efficacy our proposed method. Category-wise comparison of our approach with other methods for four emotion categories, *anger*, *fear*, *joy*, and *sadness* is presented in Fig. 1a, b, c, and d respectively. For *anger*, *fear*, and *joy* categories, our method performs better than other methods. This is due to the presence of diverse features in individual methods of the ensemble. For *sadness* category, our proposed method values are higher for *PC* and *SC* whereas CNN+SVR combination method values are slightly higher for *PCH* and *SCH*.

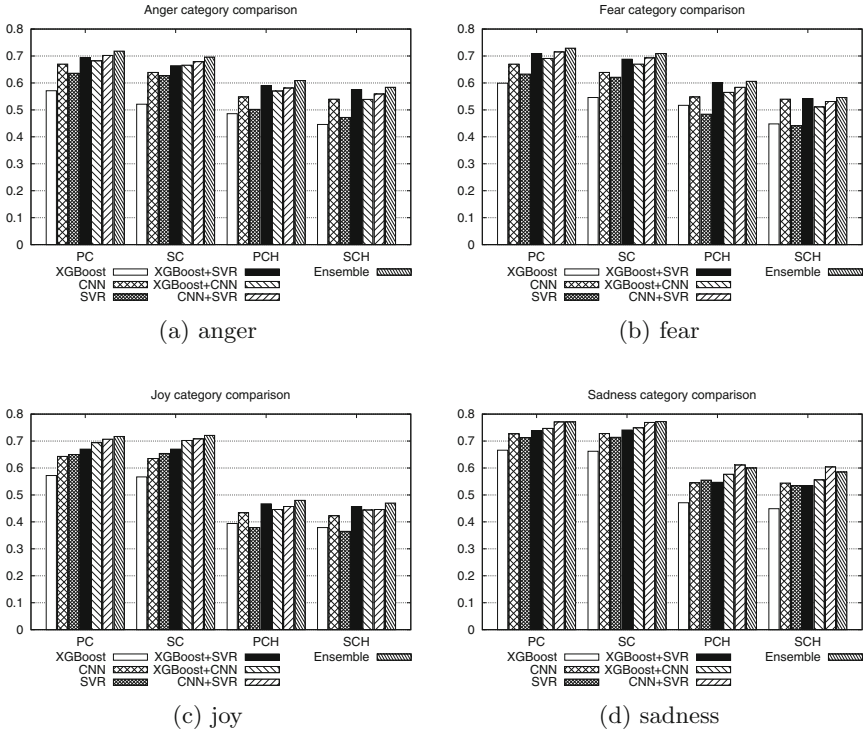


Fig. 1. Category-wise comparison of four emotion categories

6 Conclusion

In this paper, we presented an ensemble approach to predict the emotion intensity in tweets. The three methods are Convolution Neural Networks (CNN), XGBoost, and Support Vector Regression (SVR). Glove Twitter word embeddings are used with different dimensions for training the CNN model. The presence of diverse features in each of these three methods make the ensemble more stronger in predicting the better emotion intensities. Experimental results show that our method significantly outperforms other methods. For future work, we would like to identify new features and new methods to include in the ensemble.

References

1. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of twitter data. In: Proceedings of the Workshop on Languages in Social Media, pp. 30–38. Association for Computational Linguistics (2011)
2. Aman, S., Szapkowicz, S.: Identifying expressions of emotion in text. In: Matoušek, V., Mautner, P. (eds.) TSD 2007. LNCS (LNAI), vol. 4629, pp. 196–205. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74628-7_27

3. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In: LREC, vol. 10, pp. 2200–2204 (2010)
4. Bollen, J., Mao, H., Pepe, A.: Modeling public mood and emotion: twitter sentiment and socio-economic phenomena. In: ICWSM 2011, pp. 450–453 (2011)
5. Bravo-Marquez, F., Frank, E., Mohammad, S.M., Pfahringer, B.: Determining word-emotion associations from tweets by multi-label classification. In: WI 2016, pp. 536–539. IEEE Computer Society (2016)
6. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM (2016)
7. Das, D., Bandyopadhyay, S.: Identifying emotional expressions, intensities and sentence level emotion tags using a supervised framework. In: PACLIC, vol. 24, pp. 95–104 (2010)
8. Hagen, M., Potthast, M., Büchner, M., Stein, B.: Webis: an ensemble for twitter sentiment detection. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 582–589 (2015)
9. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177. ACM (2004)
10. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint [arXiv:1408.5882](https://arxiv.org/abs/1408.5882) (2014)
11. Köper, M., Kim, E., Klinger, R.: IMS at EmoInt-2017: emotion intensity prediction with affective norms, automatically extended resources and deep learning. In: Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 50–57 (2017)
12. Kouloumpis, E., Wilson, T., Moore, J.D.: Twitter sentiment analysis: the good the bad and the omg! In: ICWSM 2011, vol. 164, pp. 538–541 (2011)
13. Kumar, A., Sebastian, T.M.: Sentiment analysis on twitter. *Int. J. Comput. Sci. Issues (IJCSI)* **9**(4), 372 (2012)
14. Madisetty, S., Desarkar, M.S.: NSEmo at EmoInt-2017: an ensemble to predict emotion intensity in tweets. In: Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 219–224 (2017)
15. Mohammad, S.M.: Word affect intensities. arXiv preprint [arXiv:1704.08798](https://arxiv.org/abs/1704.08798) (2017)
16. Mohammad, S.M., Bravo-Marquez, F.: Emotion intensities in tweets. In: Proceedings of the Sixth Joint Conference on Lexical and Computational Semantics (*Sem), Vancouver, Canada (2017)
17. Mohammad, S.M., Bravo-Marquez, F.: WASSA-2017 shared task on emotion intensity. In: Proceedings of the Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA), Copenhagen, Denmark (2017)
18. Mohammad, S.M., Kiritchenko, S.: Using hashtags to capture fine emotion categories from tweets. *Comput. Intell.* **31**(2), 301–326 (2015)
19. Mohammad, S.M., Kiritchenko, S., Zhu, X.: NRC-Canada: building the state-of-the-art in sentiment analysis of tweets. arXiv preprint [arXiv:1308.6242](https://arxiv.org/abs/1308.6242) (2013)
20. Mohammad, S.M., Turney, P.D.: Crowdsourcing a word-emotion association lexicon. *Comput. Intell.* **29**(3), 436–465 (2013)
21. Niculescu-Mizil, A., Perlich, C., Swirszcz, G., Sindhwani, V., Liu, Y., Melville, P., Wang, D., Xiao, J., Hu, J., Singh, M., et al.: Winning the KDD cup orange challenge with ensemble selection. In: Proceedings of the 2009 International Conference on KDD-Cup 2009, vol. 7, pp. 23–34. JMLR.org (2009)

22. Nielsen, F.Å.: A new ANEW: evaluation of a word list for sentiment analysis in microblogs. arXiv preprint [arXiv:1103.2903](https://arxiv.org/abs/1103.2903) (2011)
23. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014). <http://www.aclweb.org/anthology/D14-1162>
24. Petrovic, S., Osborne, M., Lavrenko, V.: The Edinburgh twitter corpus. In: Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media, pp. 25–26 (2010)
25. Poursapanj, H., Weissbock, J., Inkpen, D.: uOttawa: system description for SemEval 2013 task 2 sentiment analysis in twitter. In: SemEval@NAACL-HLT, pp. 380–383 (2013)
26. Purver, M., Battersby, S.: Experimenting with distant supervision for emotion classification. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 482–491. Association for Computational Linguistics (2012)
27. Roberts, K., Roach, M.A., Johnson, J., Guthrie, J., Harabagiu, S.M.: EmpaTweet: annotating and detecting emotions on twitter. In: LREC 2012, pp. 3806–3813 (2012)
28. Saif, H., He, Y., Alani, H.: Semantic sentiment analysis of twitter. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012. LNCS, vol. 7649, pp. 508–524. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35176-1_32
29. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment strength detection for the social web. *J. Am. Soc. Inform. Sci. Technol.* **63**(1), 163–173 (2012)
30. Töscher, A., Jahrer, M., Bell, R.M.: The BigChaos solution to the Netflix grand prize. Netflix prize documentation, pp. 1–52 (2009)
31. Wang, W., Chen, L., Thirunarayan, K., Sheth, A.P.: Harnessing twitter “big data” for automatic emotion identification. In: 2012 International Conference on Privacy, Security, Risk and Trust (PASSAT) and 2012 International Conference on Social Computing (SocialCom), pp. 587–592. IEEE (2012)
32. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 347–354. Association for Computational Linguistics (2005)

A Graph-Based Frequent Sequence Mining Approach to Text Compression

C. Oswald¹, I. Ajith Kumar², J. Avinash²(✉), and B. Sivaselvan¹

¹ Department of Computer Engineering, Indian Institute of Information Technology,
Design and Manufacturing Kancheepuram, Chennai, India
{coe13d003,sivaselvanb}@iiitdm.ac.in

² Department of Computer Science and Engineering,
Sona College of Technology, Salem, India

kumarajith1996@gmail.com,avinashjeeva@hotmail.com

Abstract. A novel algorithm for mining sequential patterns using a graph and multi-layered compression is the main focus of the paper. Mining for patterns is done using a graph structure which allows fast and efficient mining of necessary frequent patterns in the text. These patterns are used with a modification of the seminal LZ78 algorithm to improve the efficiency of compression. Arithmetic coding is done on top of LZ78 to further reduce the redundancy in the text and to achieve higher rates of compression. The proposed approach has been tested with standard corpora and it shows promising results in comparison with Arithmetic coding.

Keywords: Arithmetic encoding · Graph · LZ78 · Sequence Mining
Text compression

1 Introduction

Compression is the process of minimizing the number of bits required to represent data. It is done by reducing redundant information in the data and reconstructing the compressed data to the original form when needed. Compression is generally of two types, lossy and lossless [11]. Lossy compression discards information which is not important or relevant thereby achieving more compression at the cost of data loss. JPEG, GIF, MPEG-4, H.264, AAC, MP-3 are some of the lossy compression techniques. Lossless compression allows reconstruction of data in its original form without any loss. It is used in text compression where even minimal loss of data can lead to misinterpretation of the text. Some of the lossless compression techniques are Huffman coding, Arithmetic coding, LZ77, LZ78, DEFLATE and LZMA, etc. [3, 5, 9, 12–14].

Naren Ramakrishnan quoted 5 perspectives of Data Mining and one among them is compression. Occam's Razor is employed by mining techniques to identify simple or brief patterns which can be equated to compression as data is represented in a simpler or smaller form [10]. In practice, the patterns obtained

from Data Mining can be used to compress data by using the pattern once and referring to it when it is repeated. Data mining refers to discovering information from data which are hidden, useful and non-trivial. This process is also known as Knowledge Discovery from Data(KDD) [4].

Frequent patterns are patterns that occur frequently in data. Finding frequent patterns plays an essential role in mining associations, correlations, and many other interesting relationships among data. Moreover, it helps in data classification, clustering, and other data mining tasks. Thus, frequent pattern mining has become an important data mining task and a focused theme in data mining research [4]. A pattern like “*to the*” occurring frequently in a story is a contiguous sequential pattern. Frequent Sequence Mining is used to mine contiguous frequent patterns from the text. Compression is done by using each pattern once and referring to them when they occur the next time.

A novel Graph-Based approach is employed in mining sequences of characters which are used in the compression process. The proposed approach constructs the graph in one pass of the text and mines all patterns which are necessary for compression in one pass of the graph. A pattern is considered frequent if it occurs a certain number of times in the text. This count is defined as minimum support which is represented as α . The patterns generated are used in a modified version of the seminal LZ78 algorithm [14]. The output from modified LZ78 is then subjected to Arithmetic coding, which assigns optimal bits per character based on their frequency thereby increasing compression. Our algorithm mines patterns efficiently with respect to time while achieving good compression. The rest of the paper is organized as follows. Section 2 presents related works in the fields of mining and compression. Section 3 comprises of a detailed explanation of the problem definition. The design is explained in Sect. 4. The results for various standard corpora are presented in Sect. 5. Conclusion and Future Work are discussed in Sect. 6.

2 Related Work

Huffman coding by David A. Huffman assigns prefix codes based on probability [5]. The Huffman method assigns a code with an integral number of bits to each symbol in the alphabet. Arithmetic coding solves the problem of assigning integer codes to the individual symbols by assigning one code to the whole input file [11,12]. The main disadvantage of statistical coding methods is that they do not consider the affinity between characters. Therefore it does not achieve the best possible compression. The next generation of algorithms started with the development of the Dictionary based compression methods. In 1977, Abraham Lempel and Jacob Ziv published their LZ77 algorithm, which was the first algorithm to compress data using a dynamic dictionary(sliding window). LZ77 traverses over the text and discovers patterns using a look-ahead buffer [13]. In 1978 the LZ78 algorithm was published, which is similar to LZ77 but it does not use a look-ahead buffer. It finds patterns by progressing through the text in one direction. The patterns generated by these algorithms are stored dynamically in

a dictionary, which is reconstructed when decoding [14]. The main disadvantage of the LZ78 is that it takes time to find longer sequences of characters and it also stores infrequent dictionary entries. Proceeding to the mining part, pattern mining began with the development of the apriori seminal algorithm, which generated all frequent patterns [2]. The main disadvantage is that space and time is wasted to find patterns which might not be useful. The Hash based Frequent Pattern Mining approach by Oswald et al., uses a modified Apriori algorithm which mines all patterns satisfying Apriori property [8]. These patterns are then pruned using a hash based technique to get the modified set of patterns which are then assigned codes using Huffman coding. This process consumes significant space and time, since it mines redundant patterns over several passes of the text. Our algorithm avoids the issue of time overhead by mining for patterns in a single pass of the text. Not all patterns which satisfy Apriori property are needed for compression. Only the patterns necessary for compression are mined rather than mining all patterns which satisfy Apriori property. The best features of statistical coding and dictionary based methods are combined with pattern mining to achieve higher compression.

3 Flow of Proposed Graph-Based Text Compression

Mining frequent patterns from text is significantly different from mining for patterns from transaction databases. Overlapping patterns might not be of use in text compression. Let p_1 and p_2 be two patterns. They are said to be overlapping if ' m ' consecutive characters in pattern p_1 also occurs in pattern p_2 (where $m < \text{length of } p_1 \text{ and length of } p_2$) but when they are concatenated, do not form a frequent pattern. For example, let 'we_can_' of frequency 2 and 'can_do_' of frequency 2 be frequent patterns, but 'we_can_can_do_' is not a frequent pattern. The text T is split into words. A *word* represents a sequence of characters with a whitespace suffix or a sequence of whitespaces. A *phrase* is a sequence of *words*. *Words* and their occurrences in the text are represented using a graph $G(V, E)$ where V represents the set of unique *words* $\{q_1, q_2, \dots, q_x\}$ in T and every edge carries several sequence numbers. Vertices in G are referred to as nodes. A *Seq#* (Sequence Number) represents the position in the text at which the words represented by the previous and next nodes occur together.

The list of patterns P is a set of tuples of the form $\langle \text{pattern}, \text{frequency}, \text{position_of_occurrence} \rangle$. The list *InfrequentWords* is a set of tuples of the form $\langle \text{word}, \text{value} \rangle$ where value is used to check whether that occurrence of the word is used in a pattern or not. A *Seq#* is said to be *valid* if previous, next *Seq#*s and the number itself are not *used*. CS represents the *Seq#* currently being processed in the mining phase. Non-overlapping patterns are mined from the graph in a single pass, which effectively reduces the time needed for compression. *InfrequentWords* is added to P and it is pruned to remove patterns with frequency $< \alpha$ to get P' which is used in a modified LZ78 algorithm, to represent the *LZ78_Coded_File*. Arithmetic coding is performed on the *LZ78_Coded_File* to get the final encoded file T' . Decompression is done by first decoding T' to the

LZ78_Coded_File using arithmetic coding. The *LZ78_Coded_File* is converted into the original file using LZ78 dynamic dictionary.

4 Frequent Sequence Mining-Based Text Compression Algorithm

The algorithm starts by scanning the entire text *T* and splitting it into words along with a single whitespace character as a suffix or as standalone characters, depending on whether *T* contains sequence of words or a single sequence of characters without spaces respectively. Contiguous whitespaces or newline characters are considered as words. This technique is well described with an example below (Figs. 1, 2 and 3).

Text T: “I.want_food_I.want_food_I.want____want____candy_candy_candy.”
 Spaces are represented as underscores. $\alpha = 2$

Algorithm 1. Graph_Compression

```

Input : Text T
Output: Encoded File E
Graph G ← Create_Graph(T)
P ← Mine_Patterns(G) /*
List of patterns
Modified_LZ78_Coded_File ← Encode_With_modified_LZ78(P)
Encoded_File ← Arithmetic_Compress(Modified_LZ78_Coded_File)
    
```

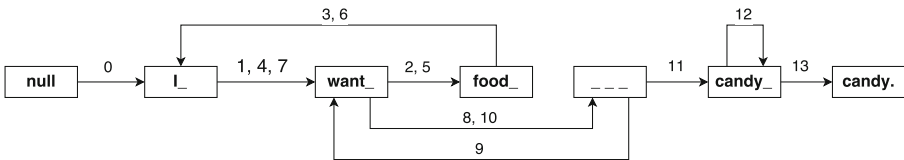


Fig. 1. Constructed graph

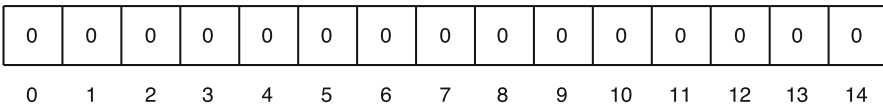


Fig. 2. Initial state of used array

In this example, *T* contains 14 words after splitting and so a boolean array *Used* of size 14 is also created. Let *Q* be the number of unique words in *T* such

0	1	1	0	1	1	0	0	1	0	1	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Fig. 3. Final state of used array

that $Q \leq W$, which is 7 in this case. Mining begins at the edge with *Seq#* 0. This edge contains one *Valid Seq#* which does not satisfy minimum support, hence this *Seq#* is not a part of any pattern. When a *Seq#* is not a part of any pattern, the next node is added to the list *InfrequentWords* as $\langle \text{Next_node}, CS + 1 \rangle$ if $CS + 1$ is unused. In this case, $\langle L, 1 \rangle$ is added. This is done to ensure that the next node has not been used in a pattern.

The edge containing sequence number 1 has three valid *Seq#*s. Hence, the phrase “I.want_” of frequency 3 is eligible to be a pattern. It is kept in memory along with the numbers 1, 4, 7, to ensure that this particular occurrence of a word is not used in more than one pattern. Since the mining of a pattern has begun, the next edge is checked to see if the pattern can be extended. In this example, if a subset of the sequence numbers 2, 5 and 8 ($1 \rightarrow 2, 4 \rightarrow 5$ and $7 \rightarrow 8$) contains at least α valid *Seq#*s and are present on the next edge and their next *Seq#*s are not used and not in memory, the pattern can be extended. 2 and 5 meet the prescribed criteria, hence the pattern in memory is extended to “I.want.food_”, thereby increasing the length of the pattern. The frequency is reduced from 3 to 2 ($1 \rightarrow 2$ and $4 \rightarrow 5$) and so the *Seq#* 7 is removed from memory and (1, 4) and (2, 5) are kept. The edge with the next *Seq#* 3 has two values 3 and 6 which are the extension of the previous values, but the value 4 is in memory. This means that we cannot extend this pattern in order to avoid overlapping patterns. The frequency falls below α and hence, the pattern $\langle \text{“I.want.food_”}, 2, (1, 4)(2, 5) \rangle$ is added to *P* and 1, 2, 4, 5 are marked as used. Every time a pattern is added to *P*, if $CS + 1$ is not used, the next node is added to *InfrequentWords*. This is done in order to avoid overlapping patterns. In this case, since 4 ($3 + 1$) is used, $\langle \text{“I_”}, 4 \rangle$ is not added to *InfrequentWords*.

The next *Seq#*s 4 and 5 are used, hence they are skipped. The edge with *Seq#* 6 has no *valid* values and since $7(6 + 1)$ has not been *used*, the next node is added to *InfrequentWords* as $\langle \text{“I_”}, 7 \rangle$. Similarly, $\langle \text{“want_”}, 8 \rangle$ is also added to *InfrequentWords*. The edge with *Seq#* 8 has two valid *Seq#*s, hence it forms a pattern. “want_” is kept in memory along with the *Seq#*s 8 and 10. The edge with *Seq#* 9 only has one valid sequence number, hence the minimum support is not satisfied. The pattern $\langle \text{“want_”}, 2, (8, 10) \rangle$ is added to *P* and 8, 10 are marked used. The next *Seq#* 10 has been used, hence we proceed to *Seq#* 11.

The edge with *Seq#* 11 does not have any valid *Seq#*s. Since 12 hasn’t been used, $\langle \text{“candy_”}, 12 \rangle$ is added to *InfrequentWords*. The edge with *Seq#* 12 is a self-loop. Whenever self loops are encountered, mining is stopped and the

pattern in memory is stored to P and the node is added to *InfrequentWords*. Since 13 has not been used, $\langle \text{"candy_"}, 13 \rangle$ is added to *InfrequentWords* and in the same way, $\langle \text{"candy_"}, 14 \rangle$ is also added to *InfrequentWords*. Tuples from *InfrequentWords* whose values have been used are removed, to ensure that the remaining words are non-overlapping. In this case, the tuple $\langle \text{"want_"}, 8 \rangle$ is removed from *InfrequentWords* since the *Seq#* 8 has been *used*. *InfrequentWords* are added to P , incrementing frequency component of that tuple everytime the same word occurs. In this case, "candy_" occurs twice, hence its frequency becomes 2.

P is pruned to remove patterns with frequency $< \alpha$. The new set of patterns is termed modified patterns P' . P' is used in the construction of tuples of the form $\langle \text{word} \rangle$, $\langle \text{dictionary_index, pattern} \rangle$ and $\langle \text{dictionary_index} \rangle$. Tuples of the second form are created once for each pattern in P when they first occur and a dictionary entry is created each time. Tuples of the third form are pointers to already present dictionary entries. The words left out after creating tuples of second and third form are made into tuples of the first form. The intermediate encoded form will be as follows.

$\langle 1, \text{"I_want_food_"} \rangle \langle 1 \rangle \langle \text{"I_"} \rangle \langle 2, \text{"want_"} \rangle \langle 2 \rangle \langle 3, \text{"candy_"} \rangle \langle 3 \rangle \langle \text{"candy_"} \rangle$

This intermediate encoded form is provided as input to arithmetic coding. Arithmetic coding considers the probabilities of all characters in the input and assigns each character a range based on its probability. These ranges are used to represent the entire file using a decimal number in the range $[0, 1)$. The final encoded file contains the frequency table and a number. Decoding is done in two phases. Arithmetic coding reconstructs the intermediate encoded form by doing the encoding process in reverse using the frequency table. The intermediate encoded form can be decoded similar to LZ78 by dynamically constructing a dictionary as follows. Tuples of the first form are directly written to the file. Tuples of the second form are added to the dictionary before writing and these dictionary entries are referenced later by tuples of the third form. In this manner, the original file is reconstructed without data loss.

5 Results and Discussions

Several experiments were conducted using "Graph-Based Arithmetic 78" (GA78) approach on several standard corpus data sets such as Canterbury, Calgary, UCI and others which covers the datasets in the range from 10 KB to 50 MB [1,6]. The Graph-Based Arithmetic 78 was implemented using java and executed on Intel(R) Core(TM) i7-6700HQ @ 2.60 GHz Notebook with 8 GB DDR4 RAM powered by Windows 10.

Table 1 helps to comparatively analyze the compression ratio and compression time with other algorithms such as Proposed Huffman 1 (FPH1), Proposed Huffman 2 (FPH2) and LZ78 [7,8,14].

The compression ratio C_r can be defined as

$$C_r = \frac{\text{Uncompressed size of Text}}{\text{Compressed size of Text}}$$

The Relative minimum support α_R is given by

$$\alpha_R = \frac{\text{Absolute Support}}{|T|} \times 100.$$

Table 1. Comparison of GA78 with other algorithms

Dataset name	α (%) at max C_r	Absolute α at max C_r	Input size T (bytes)	Compression ratio				Time (in seconds)			
				LZ78	FPH2	FPH1	GA78	LZ78	FPH2	FPH1	GA78
Canterbury											
asyoulik.txt	1.590×10^{-3}	2	125179	1.641	1.77 ($\alpha = 0.5$)	1.774	1.772	0.993	2.1	2.21	1.105
fields.c	1.790×10^{-2}	2	11150	1.333	1.38($\alpha = 0.01$)	1.384	1.557	0.58	3.95	5.26	0.174
plrabn12.txt	4.150×10^{-4}	2	481861	1.422	1.91($\alpha = 0.5$)	1.93	1.938	1.743	6.09	93.78	3.624
Calgary											
bib	1.790×10^{-3}		11261	1.728	1.99 ($\alpha = 0.2$)	1.9	1.959	0.925	1.98	12.89	0.880
book2	3.274×10^{-4}	2	610856	1.489	2.09($\alpha = 0.04$)	2.055	2.092	1.747	27.18	150.98	3.550
paper2	2.400×10^{-3}	2	82199	1.58	1.827($\alpha = 0.5$)	1.827	1.833	0.847	1.172	2.208	0.766
Large canterbury											
bible.txt	7.41×10^{-5}	3	4047392	2.061	2.62($\alpha = 0.03$)	2.549	2.656	6.232	322.35	37951.6	29.055
UCI											
UNIX_user_Data	1.760×10^{-3}	4	226867	3.228	4.48($\alpha = 0.01$)	4.489	4.714	1.106	48.18	61.97	0.917
Other											
vidb.bib	1.374×10^{-4}	2	1455480	3.296	2.75($\alpha = 0.2$)	2.419	2.873	2.18	44.88	408.94	3.142
Amazon.txt (chunked)	1.920×10^{-5}	4	20872015	2.51	-	-	2.799	29.293	-	-	248.434

(a) Minimum Support (α) vs. Compression Ratio C_r

LZ78 identifies longer patterns only when they occur several times in the text. GA78 avoids this issue by mining for patterns beforehand, thereby having a higher compression ratio than LZ78. Proposed FPHuffman 1 and 2 also mine patterns before compression, thereby having higher compression. This improvement in compression ratio can be seen for the fields.c and bible.txt. From Fig. 4 fields.c, it can be seen that compression ratio decreases with increase in α . The peak C_r is observed at $\alpha = 2$, exceeding Proposed FPHuffman 1 and Proposed FPHuffman 2 by 12%. In Fig. 4 bible.txt, the highest compression ratio for GA78 is observed at $\alpha = 3$ which surpasses Proposed FPHuffman 1 and Proposed FPHuffman 2 by a small margin and Seminal LZ78 and Huffman Coding by a large margin. Even though the number of patterns at $\alpha = 3$ is less than the number of patterns at $\alpha = 2$, the compression ratio is higher at $\alpha = 3$. This can be explained in terms of the variation in the ratio of patterns and non-patterns. These variations result in different compression ratios when they are Arithmetic coded, depending on the frequencies of characters. In general, as α increases, the number of patterns decreases, thereby leading to a decrease in compression ratio. Therefore, it can be seen that compression is higher for lower values of α . It can be inferred that α is inversely proportional to C_r .

(b) Minimum Support (α) vs. compression time

Among the algorithms compared in Table 1, it can be observed that LZ78 is faster than GA78, Proposed FPHuffman 1 and 2. This is because LZ78 compresses the

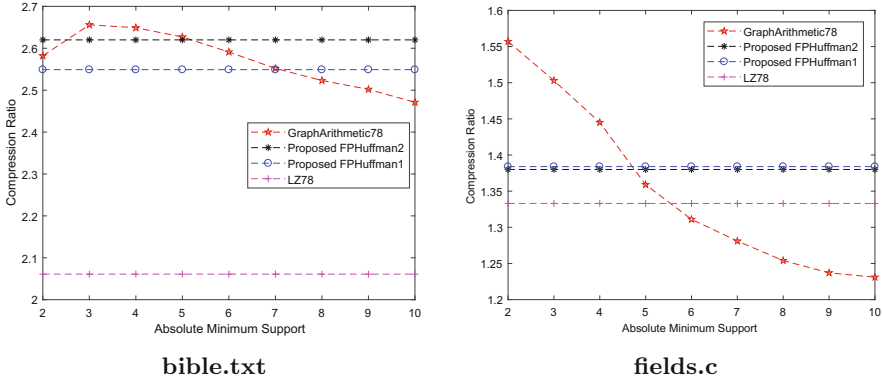


Fig. 4. α vs. C_r

text T in a single pass, however in terms of compression ratio, the rest are far superior. Between Proposed FPHuffman 1 and 2, the latter performs better in time because of the Hash data structure which helps in efficiently locating the pattern in T . Among FPHuffman 1, 2 and GA78, GA78 outperforms others in time. This can be observed in Table 1 for bible.txt where GA78 is around 10 times faster than Proposed FPHuffman 2 and around 1200 times faster than Proposed FPHuffman 1. The main reason for the tremendous difference in time is that FPHuffman 1 and 2 uses *A priori* algorithm as a base to mine patterns. FPHuffman 1 and 2 finds all possible patterns P in T which are overlapping and redundant, causing an overhead. P has to be trimmed to get the final and condensed set of patterns required for compression. GA78 on contrary does not mine overlapping and redundant patterns, but mine the patterns required only for compression “on the go” using a graph, thereby saving time several folds. In Fig. 5 bible.txt, it can be seen that the compression time is 35 seconds on average for Graph-Based Arithmetic 78. This is significantly slower than LZ78, but in terms of compression ratio, Graph-Based Arithmetic 78 is far superior. Figure 4 indicates that compression time for Graph-Based Arithmetic 78 is faster than other algorithms. From Fig. 5, it is clear that compression time remains almost the same for different α values. The compression time depends on the text T . If T has longer patterns, the compression time is longer. This is because, everytime a pattern is mined or extended, several checks are done to ensure that the pattern is non-overlapping which increases the overall compression time. If T has smaller patterns, the compression time decreases because of the decrease in the number of checks. Hence, it can be inferred that compression time is independent of α .

(c) Minimum Support (α) vs. $|P'|$

From Fig. 6 bible.txt, it can be seen that the number of patterns peaks at $\alpha = 2$. This is because when α is lower, more number of patterns in T satisfy α . However, as α increases, the number of patterns found in T decreases steadily. The same can be observed in Fig. 6 fields.c, where the patterns decrease with

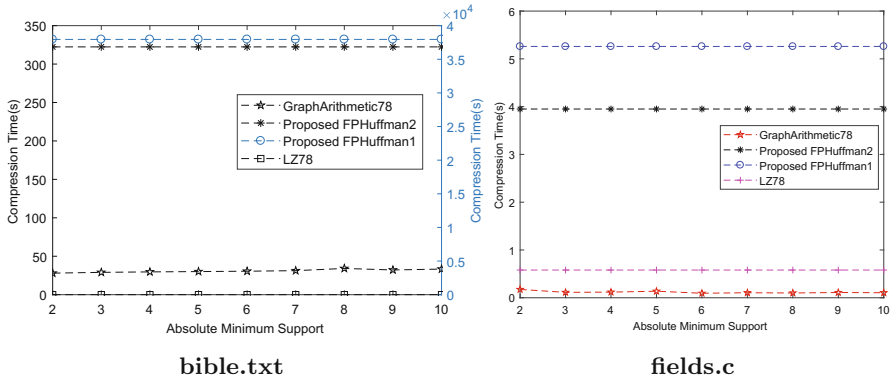


Fig. 5. α vs. time

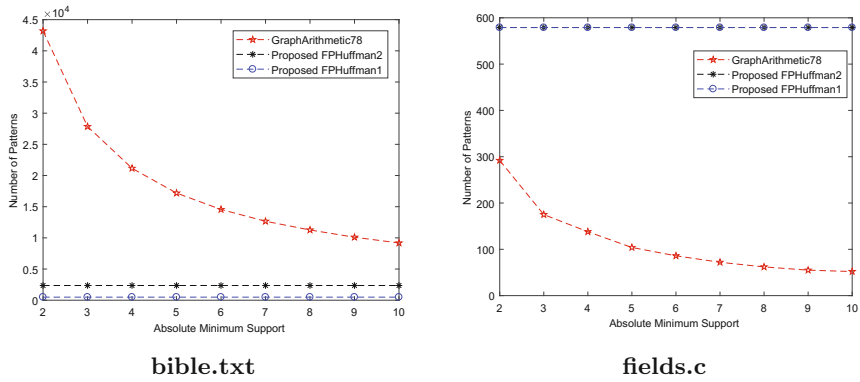


Fig. 6. α vs. $|P'|$

increase in α . The number of patterns observed in Graph-Based Arithmetic 78 is larger in comparison with Proposed FPHuffman 1 and Proposed FPHuffman 2. This results in significantly higher compression at lower α values.

6 Conclusion

The main improvement of Graph-Based Arithmetic 78 over Proposed FPHuffman 1 and Proposed FPHuffman 2 is that, there is a significant improvement in the compression time. This improvement in time is mostly due to the different mining techniques used. Apriori algorithm was used in the mining of patterns in Proposed FPHuffman 1 and Proposed FPHuffman 2. Graph-Based Arithmetic 78 uses a graph structure to efficiently patterns in a single pass, thereby effectively reducing the time taken to mine patterns. The algorithm is not only time efficient, but also space efficient as only patterns which will be used in the compression are mined. This is space efficient in comparison with Proposed huffman

algorithms as they mine all candidate patterns. The compression ratio has been observed to be higher for larger files. More observations can be made to decide the best possible way to split text in order to achieve a balance between C_r and compression time.

References

1. Calgary compression corpus datasets. corpus.canterbury.ac.nz/descriptions/. Accessed 23 July 2015
2. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: ACM SIGMOD Record, vol. 22, pp. 207–216. ACM (1993)
3. Deutsch, L.P.: Deflate compressed data format specification version 1.3 (1996)
4. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques. Elsevier, Amsterdam (2011)
5. Huffman, D.A.: A method for the construction of minimum-redundancy codes. In: Proceedings of the IRE, vol. 40, no. 9, pp. 1098–1101 (1952)
6. Lichman, M.: UCI machine learning repository (2013). <http://archive.ics.uci.edu/ml>
7. Oswald, C., Ghosh, A.I., Sivaselvan, B.: An efficient text compression algorithm - data mining perspective. In: Prasath, R., Vuppala, A.K., Kathirvalavakumar, T. (eds.) MIKE 2015. LNCS (LNAI), vol. 9468, pp. 563–575. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26832-3_53
8. Oswald, C., Sivaselvan, B.: An optimal text compression algorithm based on frequent pattern mining. J. Ambient Intell. Humaniz. Comput. 1–20 (2017). Springer
9. Pavlov, I.: LZMA SDK (software development kit) (2007)
10. Ramakrishnan, N., Grama, A.Y.: Data mining: from serendipity to science. Computer **32**(8), 34–37 (1999)
11. Salomon, D.: Data Compression: The Complete Reference. Springer Science & Business Media, Heidelberg (2004). <https://doi.org/10.1007/978-1-84628-603-2>
12. Witten, I.H., Neal, R.M., Cleary, J.G.: Arithmetic coding for data compression. Commun. ACM **30**(6), 520–540 (1987)
13. Ziv, J., Lempel, A.: A universal algorithm for sequential data compression. IEEE Trans. Inf. Theor. **23**(3), 337–343 (1977)
14. Ziv, J., Lempel, A.: Compression of individual sequences via variable-rate coding. IEEE Trans. Inf. Theor. **24**(5), 530–536 (1978)

ULR-Discr: A New Unsupervised Approach for Discretization

Habiba Drias^(✉), Nourelhouda Rehkab, and Hadjer Moulai

LRIA Laboratory, Department of Computer Science, USTHB, Algiers, Algeria
hdrias@usthb.dz

Abstract. In this work, we propose a novel unsupervised discretization method based on a Left to Right (LR) scanning technique, namely *ULR-Discr*. Its originality resides in the fact it uses fusion and division operations at the same time and among its strengths, we report two advantages. The first one consists in designing the algorithm by crossing the input stream in a single pass, and this way the time complexity is significantly reduced relatively to that of the previous works. The second is the possibility offered to provide easily any cut-point function to reach the desired effectiveness. To evaluate our method, extensive experiments were conducted on large datasets in order to undertake comparison with several classical discretization methods and recent ones.

Keywords: Data mining · Data pre-processing
Unsupervised discretization · Division and merging framework
Lexical generator

1 Introduction

The discretization of the continuous attributes helps to improve classifier performance and efficiency because the cardinality of the discrete data is smaller than that of the continuous data. Two main frameworks exist to undertake discretization: a top-down method based on a series of data division into intervals until a stop criterion is met and a bottom-up method based on a series of merging intervals until reaching the stop criterion. The main issue is to determine a cut-point in both cases and find the stop criterion to complete the process.

After sorting the values, the top-down method searches the best cut-point, with the aim to divide the whole set of data into two adjacent and distinct intervals. Then, the same process is applied and repeated to the generated intervals until a stop criterion is satisfied. Entropy, binning, correlation, precision and others are measures used to decide whether there will be a division or not. Examples of top-down algorithms are equal-width procedure where all intervals have the same width or size and equal-frequency procedure where each interval contains the same number of values.

A bottom-up discretization method starts by considering each value as a cut-point. As the process progresses, it merges the adjacent intervals that are

considered similar. An independence test such as χ^2 is used to determine whether two adjacent intervals are independent or not. If the test is positive then the process cuts the current interval and starts searching for the next one. Otherwise, it pursues the construction of the current interval.

The methods of discretization can also be classified according to the use of the class attribute. Unlike a supervised method that uses the class attribute, an unsupervised method ignores utterly the class attribute. Equal-width and equal-frequency were the first approaches developed for unsupervised discretization.

In this paper, we propose an algorithm called *ULR-Discr* for unsupervised discretization using a lexical generator [7]. The input of the generated lexical analyser is the values to be discretized. The sequence of data is read from left to right and a discretization measure is calculated between each two consecutive intervals. The values that satisfy this measure are the cut-points. A cut-point corresponds to the recognition of an interval, while the storage of a value in an interval corresponds to the construction of the interval.

The rest of the paper is organized as follows. In Sect. 2, a state-of-the-art of existing discretization methods is provided. Section 3 presents the proposed approach and explains how for the first time Lex generator is used to build a discretization algorithm. Section 4 describes three examples of statistic measures for intervals' independence, used in our experimentations. Section 5 shows the experimental results of *ULR-Discr* and compares them to the outcomes of other methods found in the literature. At the end, we conclude this work and provide some future perspectives.

2 Related Work

In the last few years, several discretization algorithms were developed. They differ from each other by the various options offered for the design: top-down or bottom-up, supervised or unsupervised and the numerous splitting and merging measures. A chronological survey of the most approaches related to our work, is presented below.

ChiMerge [6] is the first discretization method that proposes an approach other than division using the measurement of χ^2 . It is a supervised bottom-up method that exploits the relationship between the attribute and its class to merge adjacent intervals.

In [3], the authors proved that the induction algorithms are more efficient if the data are previously discretized. They tested three methods as a preprocessing step to the C4.5 algorithm [10] and a Naive-Bayes classifier: equal width Intervals, 1RD proposed by [5] and the entropy minimization heuristic devised by [4]. The results revealed that all these methods and especially the entropy heuristic, improved the Naive-Bayes classifier accuracy.

In [8] a comprehensive synthesis on discretization techniques is presented. The authors discuss the history of discretization, existing methods and their influences on classification and other applications.

An unsupervised top-down method is presented in [2]. The latter uses the estimation of the density of the nucleus to determine the cut-points. To select

the maximum number of intervals, this method uses the cross-validated log-likelihood. Experiments were performed to compare the proposed method to the equal-width and equal-frequency procedures. The results showed the superiority of this method.

The authors of [1] proposed a hybridization of chiMerge and chi2 called *CHID* and added logworth, which serves to define the meaning of a set of cut-points.

The authors of [9] used the standard deviation “z-score” for the implementation of their discretization algorithm. They performed their tests on medical datasets and results showed better accuracy compared to others methods.

These efforts on discretization revealed that the problem of discretization is an active subject that still stimulates the interest of researchers. In this paper we propose a new method of discretization, which combines the strong points of the evoked methods especially chiMerge and chi2 that have greatly influenced the studies made until now. The following section provides a description of the framework we propose.

3 ULR-Discr Framework

Among the unsupervised discretization methods widely used in machine learning are Equal-width and Equal-frequency techniques. Unfortunately, these methods are characterized by their poor performance where certain concepts become impossible to learn. In this section, we propose an approach that takes into account the notion of intervals’ independence, as does Chimerge [6].

3.1 A Brief Overview of ChiMerge and Chi2

ChiMerge [6] is precise with an intra-interval uniformity and an inter-interval difference. It merges the intervals that are judged similar using the measurement of χ^2 . It can be outlined as follows:

1. Sort the n attribute values in ascending order.
2. Consider each value in a distinct interval.
3. Calculate the value of χ^2 for all adjacent intervals.
4. Merge the interval pairs with the smallest value of χ^2 .
5. Stop the process if a predefined stopping criterion is met (such as the significance level of χ^2 , the maximum number of intervals, the maximum inconsistency, etc....) for all intervals.
6. Go to (3) otherwise.

As mentioned above, the first step of discretization consists in sorting the values of the attribute to be discretized. The choice of the sorting algorithm must be judicious because it influences the performance of the ChiMerge algorithm. The stopping criterion must satisfy a minimum probability of independence between the intervals. The threshold of χ^2 is defined according to the level of significance. A too high threshold will prolong the discretization and will generate a

few intervals, whereas a too low threshold will generate a sub-discretization and a large number of intervals. Min-intervals and max-intervals are two parameters added to avoid falling into this issue.

Without considering the initial step of sorting data, $O(n^2)$ is the worst case complexity, which corresponds to the number of times the function χ_2 is called. Let us suppose that at the beginning, we have n different values. At the first iteration, the function is called $(n - 1)$ times, then in the second iteration, it is called $(n - 2)$ times, and so on. Therefore, the number of times the function χ_2 is called in total, is equal to: $\sum_{i=1}^{i=n-1} i = \frac{(n(n-1))}{2} = \frac{(n^2-n)}{2}$.

3.2 The Proposed ULR-Discr Algorithm

To avoid repeating the calculation of the test for all intervals at each iteration, we propose the following framework:

1. Sort the attribute values in ascending order.
2. Consider each value in a distinct interval.
3. initialize I_{left} with the first input value and I_{right} with the second one.
4. while not end of input file do
 - (a) Calculate the value of χ_2 between I_{left} and I_{right} adjacent intervals.
 - (b) if the value of χ_2 is greater than the significance level of χ_2
 - i. then perform a division operation: $I_{left} = I_{right}$; $I_{right} =$ the set containing the next input value.
 - ii. else perform a merging operation: $I_{left} = I_{left} // I_{right}$; $I_{right} =$ the set containing the next input value.

It is clear that this algorithm has a $O(n)$ time complexity regardless to the data sorting instruction. It is then more efficient than any other previous discretization algorithm, thanks to the division/merge strategy.

3.3 Lex as a Tool for Numeric Discretization

The division/merge strategy we propose was implemented using a Left to Right (LR) scanning approach. The algorithm can be generated automatically by Lex [7]. The latter yields the discretization program from a source containing transition rules, which specify how to transform the source file into the target one. Transition rules respect the following syntax:

$$Regular-expression \{action\}$$

The regular expressions specify lexical entities that describe intervals in our case. An action is a set of instructions, or in other words a program fragment that will be executed each time the current interval is recognized during the process of discretization. The generated discretization program takes as input the data and the threshold (for example the significance level of χ_2) and as output, it produces a set of intervals. Concretely, the input is a sorted sequence of pairs, each pair consists of a value of type ‘real’ and its frequency in the dataset, of

type ‘integer’. The Lex program we designed is outlined in Fig. 1. *Statistics* is a function used to calculate the measure for deciding whether to cut or continue the construction of an interval. It is possible to use any statistic such as χ^2 , entropy, Pearson coefficient and Manhattan distance. To change the measure, we have simply to modify the function of the statistic and leave the rest of the program unchanged.

```

space          [\t\n]+
digit          [0-9]
integer        {digit}+
%%
{space}        {}
{integer}.{integer} {value = atof(yytext);
                }
{integer}      {frequency = atoi(yytext);
                interval2 = [(value, frequency)];
                if (statistic(interval1, interval2) >= threshold)
                    then interval1 = interval2
                    else interval1 = merge (interval1, interval2)
                }
%%
function statistic (interval1,interval2);
{
}

function merge (interval1, interval2);
{
}

main {
    interval1 = empty;
    threshold = ...;
    yylex(); }
    
```

Fig. 1. The Lex source for generating the discretization program.

4 The Used Statistics

For the experiments, we tested three statistics: the χ^2 , the distance of Manhattan and the Pearson Coefficient.

4.1 Using the χ^2 statistic

Equation (1) is used to calculate the value of χ^2 . Note that it is an adaptation of the χ^2 formula, where the attribute class is ignored.

$$\chi^2 = \sum_{i=1}^{i=m} \frac{(R_i - E)^2}{E} \quad (1)$$

where:

- m is the number of intervals to be compared (2 in our case).
- R_i is the number of values in interval i .
- E is the expected frequency calculated as: $E = n/MaxIntervals$, $MaxIntervals$ being the maximum number of intervals (fixed by the user).

4.2 Using the Distance of Manhattan

The distance of Manhattan between two objects i and j of same size n is expressed by Formula (2), where $x_{i,l}$ and $x_{j,l}$ are respectively any instance of i and j .

$$d(i, j) = \sum_{l=1}^{l=n} |x_{i,l} - x_{j,l}| \quad (2)$$

4.3 Using the Pearson Coefficient

The Pearson coefficient is used to evaluate the correlation between two numerical attributes. To calculate the Pearson coefficient between two intervals I and J , we consider the intervals instead of the attributes. The intervals' values are normalized before computing the coefficient of correlation using Eq. (3).

$$\tau_{I,J} = \frac{\sum_{k=1}^{k=p} (I_k - \bar{I})(J_k - \bar{J})}{(p-1)\sigma_I\sigma_J} \quad (3)$$

where:

p is the number of instances,

I_k is an instance of interval I ,

J_k is an instance of interval J ,

\bar{I} and \bar{J} are the respective means of I and J ,

σ_I and σ_J are the respective standard deviation of I and J .

5 Experimental Results

To prove the effectiveness of our method, we performed several experiments to compare the results with those quoted in state-of-the-art. Evaluation is based on accuracy, speed and scalability.

The Lex generator as well as the Java language were used for the implementation on a machine of Processor Intel Core i5-3317U CPU @ 1.70 GHz x 4 and a RAM of 4.00 GB under the Ubuntu 14.04 LTS 64-bit operating system.

5.1 The Tested Datasets

We considered the large datasets shown in Table 1, characterized by the number of instances and the number of attributes. For the evaluation, the k-fold cross

validation combined with the C4.5 classifier is used. The k is fixed at 10, hence the dataset will be divided into k samples, one is used for the validation and the $(k-1)$ remaining samples will form the training set. The error is computed using the C4.5 with the training set. This operation is repeated k times by selecting another validation sample each time and $(k-1)$ other samples, which were not already used as validation set. For the assessment process, the following steps are undertaken:

- Set the cut-point threshold and the maximum number of intervals.
- Divide the data in two sets: the training (9/10) and the test (1/10).
- Apply the algorithm of discretization on the training set.
- Use the cut-points obtained in the previous step to discretize the test set.
- Call the algorithm c4.5 with the training and test sets and register the error rate.
- Once the 10 iterations are performed, calculate the means of the registered error rates and running time.

Table 1. Large data sets.

Data set	# instances	# attributes
German	1000	20
Yeast	1484	8
Hypothyroid	3772	29
Abalone	4177	8
satimage	6435	36
Handwritten digits	10992	16
Letter recognition	20000	16

5.2 Experimental Results for ULR-Discr

The statistics tested for the detection of the cut-points are the χ^2 , the Pearson coefficient and the Manhattan distance for the large datasets. The results are shown respectively in the following.

ULR-Discr Using χ^2 . Table 2 exhibits the results of the execution of *ULR-Discr* using χ^2 for the large datasets.

ULR-Discr Using the Pearson Coefficient. The same experiments as those held for *ULR-Discr* using χ^2 were conducted using the coefficient of Pearson as a statistic for discretization. Table 3 shows the results for the large datasets.

Table 2. Results of *ULR-Discr* using χ^2 on the large datasets.

Dataset	Threshold	Error (%)	Time (s)
German	120	26.93	0.000401
Yeast	14	64.341	0.000230
Hypothyroid	1033	1.852	0.000617
Abalone	910	75.095	0.001404
Satimage	1300	14.304	0.001411
HandDigits	2010	6.845	0.000740
Letter	1500	17.84	0.000287

Table 3. Results of *ULR-Discr* using the Pearson coefficient on the large datasets.

Dataset	Threshold	Error (%)	Time (s)
German	500	27.969	0.0004
Yeast	4500	57.477	0.0003
Hypothyroid	150000	6.205	0.0005
Abalone	12200	75.029	0.0013
Satimage	20200	18.799	0.0012
HandDigits	94000	9.524	0.0007
Letter	9	21.551	0.0003

Table 4. Results of *ULR-Discr* using the distance of Manhattan on the large datasets.

Dataset	Threshold	Error (%)	Time (s)
German	1000	29.50	0.226
Yeast	0.1	67.53	0.143
Hypothyroid	150.0	2.222	0.605
Abalone	0.1	73.389	4.874
Satimage	1400	26.829	139.91
HandDigits	1000	10.82	0.235
Letter	10	21.45	0.235

ULR-Discr Using the Distance of Manhattan. The same experiments as those of the two previous subsection were undertaken for *ULR-Discr* using the distance of Manhattan. The results are shown in Table 4.

What we observe from this bunch of experiments is that *ULR-Discr* using χ^2 is the most effective as it computes the lowest error rate. When comparing the runtime, *ULR-Discr* using χ^2 and *ULR-Discr* using the Pearson coefficient are efficient and that *ULR-Discr* using the distance of Manhattan has to be avoided.

Table 5. Error rates yielded respectively by ULR-Discr and Zdisc on the large datasets.

Dataset	ULR-Discr			Zdisc
	χ^2	Pearson	Manhattan	
German	26.93	27.969	29.50	27.6
Yeast	64.341	57.477	67.53	51.62
Hypothyroid	1.852	6.205	2.222	6.31
Abalone	75.095	75.029	73.389	46.19
Satimage	14.304	18.799	26.829	8.35
HandWritten Digits	6.845	9.524	10.82	6.87
Letter	17.84	21.551	21.45	39.05

Table 6. Comparison of runtimes yielded respectively by *ULR-Discr* and *Zdisc* on large datasets.

Dataset	ULR-Discr			Zdisc
	χ^2	Pearson	Manhattan	
German	0.000401	0.0004	0.226	0.0206
Yeast	0.000230	0.0003	0.143	0.0220
Hypothyroid	0.000617	0.0005	0.605	0.0620
Abalone	0.001404	0.0013	4.874	0.872
Satimage	0.001411	0.0012	139.91	0.5906
HandDigits	0.000740	0.0007	0.235	0.8734
Letter	0.000287	0.0003	0.235	0.0650

5.3 Comparing Different Variants of *ULR-Discr* with *ZDisc*

ZDisc [9] is an unsupervised method of discretization based on z-score normalization. The main steps of this method are:

1. Select the continuous intervals in the dataset.
2. Normalize the values of each interval using the z-score.
3. Determine the minimum ‘a’ and maximum ‘b’ after normalization, of each attribute.
4. Divide the interval [a, b) into k Equal width intervals or bins, where $(b - a)/k$ is the width or size of the intervals.

Table 5 reports the error rates calculated for the three variants of *ULR-Discr* and *Zdisc*. It shows that *ULR-Discr* using χ^2 gives the best results for large datasets, where the number of instances with the smallest error rate is the largest. Table 6 exposes the runtimes of the tested algorithms and shows the superiority of *ULR-Discr* relatively to *Zdisc*, especially when we use the Pearson coefficient and χ^2 as a discretization measure.

In this section, we presented the empirical results obtained by *ULR-Discr* on the tested large datasets. In comparison with *Zdisc*, it can be seen that *ULR-Discr* is more effective and more efficient than *Zdisc*.

6 Conclusion

In this paper, we presented a novel unsupervised method of discretization based on a left to right scanning approach. The new framework of discretization presents the following advantages:

- This framework allows processing discretization in only one pass and hence reduces the time complexity to $O(n)$, where n is the number of instances.
- For the first time, a discretization framework based on division and merging operation at the same time is designed.
- The framework allows to modify the discretization statistic with ease, thanks to the lexical generator.

The experiments showed that the results are very satisfactory for all the measures exploited, compared to the recent *ZDisc* algorithm.

References

1. Bettinger, R.: ChiD. A χ^2 -based discretization algorithm. In: Proceedings of WUSS, Modern Analytics, San Francisco, CA (2011)
2. Biba, M., Esposito, F., Ferilli, S., Di Mauro N., Basile, T.: Unsupervised discretization using kernel density estimation. In: IJCAI, pp. 696–701 (2007)
3. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: ICML 1995, pp. 194–202 (1995)
4. Fayyad, U., Irani, K.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of Thirteenth International Joint Conference on Artificial Intelligence, pp. 1022–1027. Morgan Kaufmann, San Mateo (1993)
5. Holte, R.C.: Very simple classification rules perform well on most commonly used datasets. *Mach. Learn.* **11**, 63–91 (1993)
6. Kerber, R.: ChiMerge: discretization of numeric attributes. In: AAAI Proceedings, pp. 123–128 (1992)
7. Lesk, M.E., Schmidt, E.: Lex: a lexical analyzer generator. Accessed 12 Dec 2016
8. Liu, H., et al.: Discretization: an enabling technique. *Data Min. Knowl. Disc.* **6**, 393–423 (2002). Kluwer Academic Publishers
9. Madhu, G., Rajinikanth, T.V., Govardhan, A.: Improve the classifier accuracy for continuous attributes in biomedical datasets using a new discretization method. In: 2nd International Conference on Information Technology and Quantitative Management, ITQM (2014)
10. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993)

Identifying Terrorist Index (T^+) for Ranking Homogeneous Twitter Users and Groups by Employing Citation Parameters and Vulnerability Lexicon

Soumyadeep Debnath¹, Dipankar Das^{2(✉)}, and Bappaditya Das¹

¹ Department of Computer Science and Engineering,
Jalpaiguri Government Engineering College, Jalpaiguri, India
soumyadebnath13@gmail.com, bappadityadas96@gmail.com

² Department of Computer Science and Engineering, Jadavpur University,
Kolkata, India
dipankar.dipnil2005@gmail.com

Abstract. Twitter, one of the important social media also acts as substrate to mix enzymes of actions and reactions of public sentiments including terrorist activities, either implicitly or explicitly. The identification of terrorist or crime related texts from Twitter is difficult because of the presence of such implicit and implied code-mixed hints that are solely shared among different users within a few groups. In the present task, we have developed a framework that deals with Twitter data and provides varieties of dataset useful for tracking various terrorist activities. The system also identifies the terrorist indices of different users as well as their groups. Compared to the available citation indices (e.g., *h*-index or *i10*-index), this newly proposed index takes into account the normalized parameters of citation and seed as well as code words related to crime from vulnerability lexicon. Finally, this index is used for ranking the Twitter users and groups and it is observed that our proposed ranking algorithms have performed reasonably better when the algorithms encapsulate the roles of citation and vulnerability parameters together.

Keywords: Twitter · Homogeneous group · Citation · Terrorist index
Vulnerability lexicon · Ranking

1 Introduction

Social media specially Twitter, Facebook, YouTube are being considered as a platform for common people to share information and express their opinions about current events. But, security, now a days is associated with the dynamics of the crowd in social media rather than usual websites [1]. With the rapid growth of social media, like different communities, terror links are also formed and it is beyond the scope of the security agencies to inspect all the messages exchanged/views posted in social media for potential security threats. Thus, informative social-media activities deserve to be transferred securely as they may contain potential threats and hints of terrorism. So, our primary

objective is to automatically identify threatening messages concerned to security agencies and secondary objective is terrorist profiling, i.e. to identify persons with potential terror activities based on the social media communication.

Work presented in this paper primarily focuses on Twitter and studies have been explored and highlighted the role of Twitter as a news media and a platform to gauge public sentiments. The terrorist activities happened in real world has an impact on social media as Twitter¹. Sometimes, they used to form groups for sharing information related to recent terrorist events under the umbrella of social network. Since conversations of these groups has an impact of actual terrorist attack involving their members and/or Twitter users, we used to rank groups along with the ranking of group members in order to know how much they are involved or not in the attack or in the terrorist activity.

In the present attempt, we have explored methodologies to identify whether any homogeneous group use any code-word related to any terrorist activity or not and if so, how to analyze the groups by assigning ranks according to their vulnerability status. One of our prime objectives is to identify terrorist index (T^+ index) of homogeneous groups and their members in order to rank them.

In rest of the paper, we have discussed related research work on the topic followed by details on dataset preparation. In next sections, we discuss about methodology and then, we achieved some results and analyzed the results followed by conclusion and future work.

2 Related Work

Ranking of tweets by considering the relevance and credibility is researched extensively. Credibility analysis of ranking tweets has been attempted in [2] where the authors tried to assess the credibility of individual tweets. In their analysis, they identified the important content and sourced based features, which can predict the credibility information in a tweet. Ranking tweets by considering trust and relevance has been attempted in [3] where the analysis of trustworthiness and popularity exploits the implicit relationships between the tweets. Their preliminary evaluations show improvement in precision and trustworthiness over the baseline methods and acceptable computation timings. Another path was ranking of tweets based on heterogeneous networks. Similar work has been attempted in [4] by Huang and others to construct heterogeneous networks by harnessing cross-genre linkages between tweets and semantically-related web documents from formal genres, and inferring implicit links between tweets and users. To rank tweets effectively, they introduce Tri-HITS, a model to iteratively propagate ranking scores across heterogeneous networks.

So far, the work done over Twitter is to rank tweets, but our work differs because their analysis was solely based on ranking of tweets by considering relevance, credibility or heterogeneous networks, while we focus on ranking homogeneous Twitter users and groups by using citation indices and vulnerability lexicon.

¹ <https://twitter.com/>.

3 Dataset Preparation

We have collected Twitter data in different stages in order to achieve different goals and finally prepared five important classified sets of the actual dataset sequentially. The steps are briefly described and shown in Fig. 1.

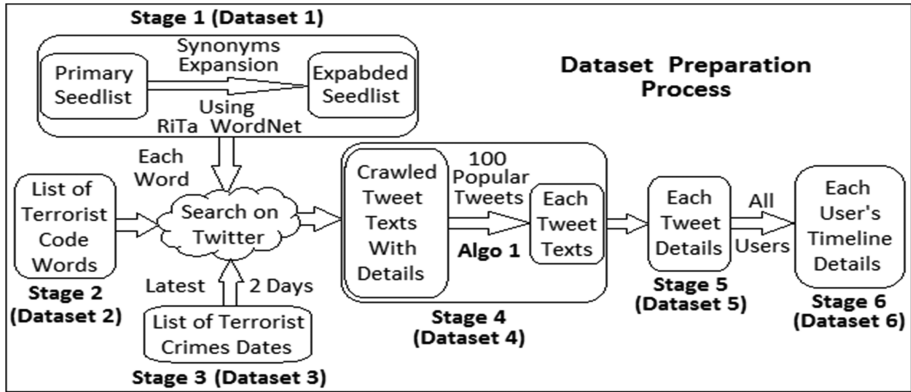


Fig. 1. Overview of data collection process

3.1 Stage 1: Preparation of List of Seed Words (*Seedlist*)

This is the preliminary part of our dataset. Observing several real incidents in the WWW, we manually prepared a primary seedlist of total 594 words related to terrorist and crimes from English vocabulary^{2,3}. In order to expand our seedlist, we collected minimum five synonyms of each word of the primary seedlist using ‘WordNet’⁴ and added them to create an expanded seedlist (**Dataset 1**) which contains 2970 seed words. We used ‘RiTaWordNet’⁵ to identify the synonyms because ‘RiTa’ is integrated with the WordNet database. Finally, we have searched each of the words in the seedlist and extracted relevant tweets related to terrorism or crime.

3.2 Stage 2: List of Terrorist Code Words (*Codelist*)

It is almost known to all of us that the Twitter users associated with highly terrorist activity use one or more than one code words during their tweet conversations in order to continue an encrypted communication within public conversations. So, from several news forecasting and latest updates, we manually created another list of total

² <https://myvocabulary.com/word-list/terrorism-vocabulary/>.

³ <https://gist.github.com/jm3/2815378>.

⁴ <https://wordnet.princeton.edu/>.

⁵ <https://rednoise.org/rita/>.

53 terrorist code words⁶ (**Dataset 2**). The code words are classified into three types long with their frequency i.e. *unigram* (23), *bigram* (11) and *trigram* (9). We also used each word from this dataset, to search relevant tweet texts.

3.3 Stage 3: Preparation of List of Current Terrorist Crimes Dates (*Crimelist*)

In order to collect strong viral data from Twitter further, we needed all types of information related to current terrorist incidents. So, we prepared a large list of total 618 dates related to terrorism crimes from several reports and considered latest 2 days duration with respect to each of the terrorist incidents for collect tweets. Merging all these dates, we prepared a list of dates⁷ (**Dataset 3**) and used each date from the dataset to search relevant viral data related with terrorist crimes from Twitter on that date.

3.4 Stage 4: Dataset of Crawled Tweet Texts with Details

We crawled a huge amount of data from Twitter by using the words from the expanded *seedlist* (**Dataset 1**) and all the time intervals of the *crimelist* (**Dataset 3**). We collected 3226 number of tweets for 968 users and the average number of tweets per user is ~ 3 . Each instance contains the tweet text with it's user name, user id, retweet count, favorite count, time, tweet id and language details. It was assured that each tweet text contains at least one word from the updated *seedlist* and the time of each tweet must belong to any one of the time intervals as mentioned in the *crimelist*. We have mentioned one sample tweet in Fig. 2.

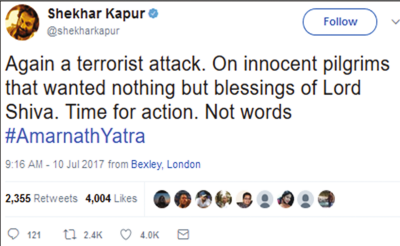
<p>User Name : Shekhar Kapur User Id : @shekharkapur Twit : Again a terrorist attack. On innocent pilgrims that wanted nothing but blessings of Lord Shiva. Time for action. Not words #AmarnathYatra Time : Mon Jul 10 09:16:04 PDT 2017 Language : en Favorite Counts : 4004 Retweet Counts : 2355 Twit Id : 884860107653120000</p>	
--	---

Fig. 2. Snapshot of a tweet data crawled from Twitter

In order to crawl the previously described data from Twitter, we used a java library of Twitter called '*Twitter4J*'⁸. For further work, we needed to construct a small amount of highly rated data from this huge dataset as input and collected top 100 most popular tweets from this dataset. To evaluate the popularity, we consider two parameters; they

⁶ <https://m.rediff.com/news/report/11-code-words-terrorists-in-india-use-a-lot/20140224.htm>.

⁷ <https://storymaps.esri.com/stories/terrorist-attacks/?year=2017>.

⁸ <https://Twitter4j.org/en/>.

are the ‘Retweet Count’ and the ‘Favorite Count’. We applied the following algorithm (Algorithm 1) to get top 100 most popular tweets (**Dataset 4**).

Algorithm 1:

Step 1: Sort the tweets in descending order in the way given below.

Step 1.1: Check the ‘Retweet Counts’ of the tweets as the first parameter for sorting.

Step 1.2: If ‘Retweet Counts’ are same for more than 1 tweet.

Step 1.2.1: Check the ‘Favorite Counts’ of those tweets as the second parameter for sort.

Step 2: Collect top 100 tweets from the sorted result.

3.5 Stage 5: Dataset of Each Specific Tweet’s Conversation Details

From the previously prepared dataset, we selected each tweet id of the 100 tweets as input and crawled all the conversation details with respect to each tweet. A total of 100 conversations contain all reply tweets, all re-reply tweets, reply user names, reply user ids and also re-reply user names, re-reply user ids etc.

Thus, we finally prepared the dataset of each specific conversation details for 100 tweets (**Dataset 5**). In these total 100 Twitter conversations, the number users in each of the conversation is on average is ~ 10 and we considered each tweet conversation as a homogeneous group associated with terrorist activities.

Twitter4j library doesn’t support any function to reach the reply conversations of a particular tweet, directly. So, to crawl the previously described data from Twitter, we used a very well-known process of web crawling, called ‘Web Scraping’. For it, we used a java library called ‘*jsoup*’⁹ and parse the HTML from a URL.

3.6 Stage 6: Dataset of Each Specific User’s Timeline Details

From the previously generated dataset, we selected each user id of all the reply users and re-reply users as input to crawl all the timeline tweets and retweet of all the users. In the dataset for each user, there are all tweets, retweets, total number of tweets, total retweet counts and retweet count of each tweet for that specific user’s timeline (**Dataset 6**). The total number of users is 1000 whereas number of tweets in each user’s timeline is on average ~ 100 and we considered each user as a member of the homogeneous group related with terrorist activities.

4 Methodology

In order to generate a new optimized ranking algorithm for the Twitter users and any type of homogeneous groups, we used our classified dataset independently by the following five sequential working stages. These stages are briefly described below (Fig. 3).

⁹ <https://jsoup.org/>.

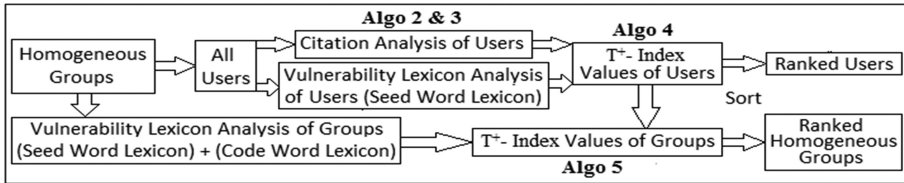


Fig. 3. Overview of experiment process

4.1 Homogeneous Group Formation

At first, we marked the homogeneous groups from Twitter data on which we had to apply our ranking algorithm. Therefore, we used the dataset of each specific tweet’s conversation details (i.e., **Dataset 5**) to create the homogeneous groups. In each group (conversation) users who are involved within the same conversation with respect to tweet topic. Thus, we considered the whole conversation group for each tweet as a homogeneous group and finally we obtained dataset of total 100 homogeneous groups where each group contain ~ 10 average number of users.

4.2 Citation Analysis of Twitter Users

This working stage was the first stage for our ranking approach. In order to evaluate the normalized values for citation analysis, we calculated four parameters for each of the users; total citation count, total number of tweets, *h*-index and *i10*-index.

In order to calculate these parameters, we use the dataset of each specific user’s timeline details (i.e., **Dataset 6**). From this dataset, we gathered total number of users, total number of tweets of each user and all the retweet counts of all tweets for each user. The second parameter was already calculated.

4.2.1 Counts of Total Citations

We applied this algorithm (Algorithm 2) to get the total citation counts of all Twitter users of all homogeneous groups.

Algorithm 2:

Step 1: Select any of the users from total number of users.

Step 1.1: Calculate citation counts of total number of tweets of that user.

Step 1.1.1: Select any tweet of that user.

Step 1.1.2: Count citation of any tweet = retweet count of that tweet (using formula).

Step 1.1.3: Repeat the same process for all other tweets.

Step 1.2: Calculate total citation count of that user.

Step 1.2.1: Total citation count = sum of all retweet counts of all tweets.

Step 2: Repeat the same process for all other users.

4.2.2 Identification of Citation Indices (*h*-Index and *i10*-Index)

We all are familiar with the algorithm for finding the *h*-index and *i10*-index of any ‘Google Scholar’ profile. So, we implemented the similar algorithm¹⁰ (Algorithm 3), given below for finding the *h*-indices and *i10*-indices of all the Twitter users.

Algorithm 3:

Step 1: Select any of the users from total number of users.

Step 1.1: Achieved citation count of each tweet for all tweets of that user.

Step 1.2: H-Index = number of tweets (h) with a retweet count (citation number) $\geq h$.

Step 1.3: i10-Index = the number of tweets with at least 10 retweet counts (citations).

Step 2: Repeat the same process for all other users.

4.3 Preparation of Vulnerability Lexicon

The lexicon related to vulnerability is one of the important contributions of our present ranking approach for analysis of various Twitter users and their homogeneous groups. In order to evaluate or analyze lexicons, we needed to deduce the normalized values in two different ways; seed words and code words. We used the normalized values of only seed words for ranking the users while for ranking the groups we used the normalized values of both the seed words and code words.

4.3.1 Seed Word Lexicon Analysis

Here, we assumed two parameters for a tweet conversation; total number of different words from seedlist (**Dataset 1**) and total number of times each word is appeared. Thus, to calculate the parameters, we used two different input datasets for users and homogeneous groups and they are; for users’ seed word lexicon, all tweet texts from the dataset of each specific user’s timeline details (i.e., **Dataset 6**) whereas for homogeneous groups’ seed word lexicon, all tweet texts from the dataset of each specific tweet’s conversation details (i.e., **Dataset 5**). We also used *Porter Stemmer*¹¹ to stem the words in order to obtain more relevant instances from these datasets and calculated the counts of all such seed words having several prefixes and suffixes.

4.3.2 Code Word Lexicon Analysis

In this case, we also assumed similar two parameters for a tweet conversation; total number of different words from codelist (**Dataset 1**) and total number of times each word is used. Thus, to calculate these parameters, we used all tweet texts from the dataset of each specific tweet’s conversation details (i.e., **Dataset 5**) for code word lexicon of homogeneous groups. Here, we used three types of n-grams for searching; unigram, bigram and trigram, respectively.

¹⁰ <http://libguides.cmich.edu/c.php?g=353669&p=2468209>.

¹¹ <https://github.com/stanfordnlp/CoreNLP/blob/master/src/edu/stanford/nlp/process/Stemmer.java>.

4.3.3 Parts-of-Speech (POS) Tags

We have analyzed all the seed words and code words from tweet texts to recognize their parts of speech of eight different types (*noun, pronoun, adjective, verb, adverb, preposition, conjunction and interjection*). Here, we used the java library of ‘*Ark parts of speech (POS) tagger*’¹². From the statistical details of this lexicon analysis (mentioned in Table 1), it is reflected that in Twitter the terrorist activities are mainly focused by the noun words.

4.3.4 Named Entity Tags

Not only POS tags, we also analyze all the seed words and code words from tweet texts to recognize the name-entity fields of seven different types (*time, location, organization, person, money, percent and date*). Here, we used the java library of ‘*Stanford Named Entity Recognizer (NER)*’¹³ containing 7 classifiers which label sequences of words in a text with 7 types. From the statistical details of this lexicon analysis (mentioned in Table 1), we conclude that in Twitter, terrorist activities are mainly focused on words which are not named entities.

Table 1. Parts of Speech (POS) and Name Entity (NE) statistics of seed and code words

Types	Seed words (unigram)	Code words		
		Unigram	Bigram	Trigram
<i>Part of Speech (POS)</i>	<i># Parts of Speech (POS)</i>			
Noun	388	23	21	23
Adjective	86	0	1	1
Verb	97	0	0	0
Adverb	41	0	0	0
Conjunction	0	0	0	3
<i>Name Entities (NEs)</i>	<i># major Name Entity</i>			
Person	6	1	2	0
Location	4	0	1	1
Organization	11	1	0	0

4.4 Ranking of Twitter Users

Here, each user is being assigned with a parametric value (T^+ -index), known as T_u^+ -index and based on such indices, we have arranged them in decreasing order of their vulnerability. Thus, we complete the ranking procedure with respect to all the users of all homogeneous groups. In order to obtain this parametric value for each user, we used two types of normalized values i.e., citation analysis values and vulnerability lexicon values, both. We have described our algorithm (Algorithm 4) for ranking homogeneous Twitter users below.

¹² <https://github.com/vatsan/gp-ark-tweet-nlp>

¹³ <http://guides.library.cornell.edu/c.php?g=32272&p=203388>.

4.5 Ranking of Homogeneous Twitter Groups

Here, each group is being assigned with a parametric value (T^+ -index), known as T_g^+ - index and based on such indices, we have arranged them in decreasing order of their vulnerability. Thus, we complete the ranking procedure with respect to all the users of all homogeneous groups. In order to obtain this parametric value for each group, we used two types of normalized values i.e., users' parametric values and vulnerability lexicon values, both. Now, we have described our algorithm (Algorithm 5) for ranking homogeneous Twitter groups below.

Algorithm 4:

Total number of users = N . User count, $u = 1$ to N . Total number of tweets of any user = T_u . Total citation count of any user = C_u . H-Index of any user = H_u . I^{10} -Index of any user = I^{10}_u . Number of different seed words from seedlist in any user's timeline = $W_s N_u$. Number of times those seed words are used in any user's timeline = $W_s T_u$. Normalized value of citation analysis of any user = NC_u . Normalized value of vulnerability lexicon analysis of any user = NL_u . T^+ -index value of any user = T^+_u

Step 1: Select any of the users from total number of users.

Step 1.1: Calculate the normalized value of citation analysis of that user.

Step 1.1.1: $NC_u = (H_u + I^{10}_u) * (C_u / T_u)$

Step 1.2: Calculate the normalized value of vulnerability lexicon analysis of that user.

Step 1.2.1: $NL_u = (W_s N_u * 1) + (W_s T_u * 0.5)$

Step 1.3, Calculate the T^+ -index value of any user.

Step 1.3.1: $T^+_u = (NC_u + NL_u) / 1000$

Step 2: Repeat the same process for all other users.

Step 3: Sort T^+_u of all users in decreasing order.

Algorithm 5:

Total number of homogeneous groups = N . Homogeneous group count, $g = 1$ to N . Total number of users in any homogeneous groups = n_g . User count of any homogeneous group, $u_g = 1$ to n_g . T^+ -index value of any user = $T^+_{u_g}$. Number of different seed words from seedlist in any homogeneous group's conversation = $W_s N_g$. Number of times the seed words are used in any homogeneous group's conversation = $W_s T_g$. Number of different code words from the list of terrorist code words in any homogeneous group's conversation = $W_c N_g$. Number of times the code words are used in any homogeneous group's conversation = $W_c T_g$.

Normalized value of user's T^+ -index to analyze any homogeneous group = NP_g . Normalized value of vulnerability lexicon of any homogeneous group = NL_g . T^+ -index value of any homogeneous group = T^+_g

Step 1: Select any of the homogeneous groups from total number of homogeneous groups.

Step 1.1: Calculate normalized user's parametric values of that homogeneous group.

Step 1.1.1: $NP_g = (\sum P_{u_g}) / n_g$, for $u_g = 1$ to n_g

Step 1.2: Calculate normalized value of vulnerability lexicon of that homogeneous group.

Step 1.2.1: $NL_g = ((W_s N_g * 1) + (W_s T_g * 0.5)) + (((W_s N_g * 1) + (W_s T_g * 0.5)) * 2)$

Step 1.3: Calculate the T^+ -index value of any group.

Step 1.3.1: $T^+_g = (NP_g + NL_g) / 1000$

Step 2: Repeat the same process for all other homogeneous groups.

Step 3: Sort T^+_g of all homogeneous groups in decreasing order.

5 Results and Analysis

We analyzed the whole outputs in two subsequent categories; these are briefly described below.

5.1 Analysis of Users' Ranking

For each Twitter user, two main parameters of the T^+ -index (i.e., T_u^+ -index) basically represent two aspects; the normalized citation value represents the popularity of that user in Twitter whereas the normalized vulnerability lexicon value represents the involvements of that user with respect to terrorism in Twitter.

By using these two parameters as axes, we have plotted a graph for 10 users of a homogeneous group and we observed that the user having both values as maximum achieves the top rank. But for the same users, when we have replaced the vertical axis with three different values; *h*-indices, *i10*-indices, the values of the (*total citation counts/total tweet counts*) and the normalized values of vulnerability lexicon, then the graph has been turned into non-symmetric and also easily not acceptable for users' ranking. It justifies that our implemented algorithm results positive and efficient output for users' ranking (shown in Fig. 4).

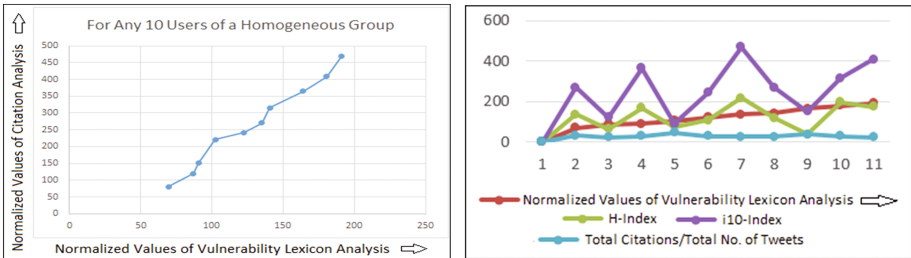


Fig. 4. Analysis of user's ranking

5.2 Analysis of Homogeneous Groups' Ranking

It is true that the homogeneous Twitter groups can be mutually exclusive. It means there can be one or more than one user belong to more than one group. For each group, two main parameters of the T^+ -index (i.e., T_g^+ -index) basically represent two aspects; the normalized users' terrorist index represents the popularity of that group in Twitter whereas the normalized vulnerability lexicon value represents the involvements of that group with respect to terrorism in Twitter.

By using these two parameters as axes, we have plotted a graph for 10 groups as shown in Fig. 4 and we can observe that the group having both values maximum obtains the top rank. Therefore, it justifies that our implemented algorithm results positive and efficient output for group ranking (shown in Fig. 5).

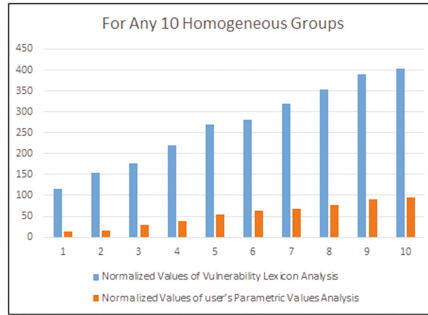


Fig. 5. Analysis of group's ranking

6 Conclusion and Future Work

At the end, we can conclude that we have generated a new algorithmic approach for ranking any number of homogeneous groups with respect to their terrorist activities in Twitter by analyzing both vulnerability lexicon and group user's ranking. In order to do this, we have also generated another new algorithmic approach for ranking any number of users according to their terrorist involvements in Twitter.

So, by using these ranking algorithms of Twitter users and homogeneous groups, anyone can also proceed this research work further more. Such as, if anyone can create any number of groups using different type of Twitter users according to own necessity, then these algorithms can be implemented for the ranking of any type of heterogeneous Twitter groups also.

References

1. Cambria, E., Rajagopal, D., Olsher, D., Das, D.: Big social data analysis. In: Akerkar, R., (ed.) Big Data Computing, Chap. 13, pp. 401–414. Taylor & Francis Group/CRC, Boca Raton (2015). Hardback: 978-1-46-657837-1, eBook: 978-1-46-657838-8.2013
2. Gupta, A., Kumaraguru, P.: Credibility ranking of tweets during high impact events. In: Proceedings of the 1st Workshop on Privacy and Security in Online Social Media, 17 April 2012, p. 2. ACM (2012)
3. Ravikumar, S., Balakrishnan, R., Kambhampati, S.: Ranking tweets considering trust and relevance. In: Proceedings of the Ninth International Workshop on Information Integration on the Web, 20 May 2012, p. 4. ACM (2012)
4. Huang, H., Zubiaga, A., Ji, H., Deng, H., Wang, D., Le, H.K., Abdelzaher, T.F., Han, J., Leung, A., Hancock, J.P., Voss, C.R.: Tweet ranking based on heterogeneous networks. In: COLING 2012, pp. 1239–1256 (2012)

Soft Metaphor Detection Using Fuzzy c-Means

Sunny Rai¹(✉), Shampa Chakraverty¹, Devendra K. Tayal², and Yash Kukreti¹

¹ Division of Computer Engineering, NSIT, Delhi, India
post2srai@gmail.com, apmahs.nsit@gmail.com, yashkukreti8117@gmail.com

² Department of Computer Science and Engineering, IGDTUW, Delhi, India
dev_tayal2001@yahoo.com

Abstract. Prior works in metaphor detection have largely focused on crisp binary classification of textual input into ‘metaphorical’ or ‘literal’ phrases. However, the journey of a metaphor from being *novel* when newly created to eventually being considered *dead* due to the acquired familiarity with the mapping over a time span, is a continuum. This observation guides us to the idea that a metaphorical text is indeed, partially literal and partially metaphorical. In this paper, we investigate the idea of soft metaphor detection by assigning membership values to fuzzy sets representing varying degrees of metaphoricity. We use a set of conceptual features and apply a simple unsupervised technique of Fuzzy c-means to illustrate fuzzy nature of metaphors. We report our experimental results on a dataset of nominal metaphors to illustrate the concept of soft metaphor detection and demonstrate their simultaneous membership in multiple classes by visualizing overlapping clusters, *metaphor* and *literal*.

1 Introduction

A metaphor is a cognitive phenomenon which maps an abstract concept in a target domain to a relatively concrete concept from a well-defined source domain to ease the understandability in communication [7]. The underlying idea is to discover patterns in the learned source domain which somehow helps in illustrating the abstract concept. The selected mapping *i.e.* metaphor also reflects the perception of a writer about the target concept. Let us consider a sentence:

An atom is a solar system. (a)

Here, the mapping is performed between the *solar system* (source domain) and *atom* (target domain). In this mapping, tiny particles such as *protons* revolving around the nucleus are analogous to *planets* and *nucleus* corresponds to *sun*. To comprehend the sentence (a), we import our existing knowledge about structure of the *solar system* such as ‘*planets revolve around the sun*’, ‘*a planet have an orbit*’ to understand the model of an *atom*.

In the late 1980s, Nunberg categorized metaphors into two categories namely *dead metaphors* and *novel metaphors* [12]. As the terminology implies, dead

metaphors are overly exploited mappings in daily parlance and thus adopt the metaphorical meaning as an extended literal interpretation. One such example is the usage of the word *gem* in the phrase ‘My husband is a gem.’ The word, *gem*¹ initially meant ‘*a crystalline rock that can be cut and polished for jewelry*’ or ‘*a precious or semiprecious stone incorporated into a piece of jewelry*’ but at present, its meaning extends to ‘*a person who is as brilliant and precious as a piece of jewelry*’ which is a metaphorical interpretation. Dead metaphors are considered equivalent to literal text as one does not need to perform any mental mapping to interpret it. In contrast, novel metaphors are newly generated mappings and require common-sensical knowledge of concepts involved in the mapping to extract contextually coherent metaphorical interpretation.

Recent studies in metaphor detection have largely focused on crisp binary classification of textual input as metaphorical or literal [6, 11, 14, 17]. However, a metaphor undergoes a process of gradual transition from being initially *novel* to eventually being considered *dead*. The transition begins from the state of highest metaphoricality when a metaphor is created and used for the first time and gradually declines to the lowest metaphoricality when it is considered to be a dead metaphor due to its frequent usage in common parlance. Thus, we argue that a hard classification of a given phrase exclusively to the classes *metaphor* or *literal* is not a pragmatic approach.

The extent of novelty or metaphoricality of a metaphor is determined by the uniqueness of the mapping between the source domain and the target domain concepts in a metaphorical expression. Working on the hypothesis that a novel metaphor is generated when an unseen or rare comparison is made between two unrelated concepts, we infer the extent of novelty of a metaphorical expression by examining the semantic relatedness between the mapped concepts. Semantic relatedness is a metric based on distributional hypothesis which states “words which are similar in meaning occur in similar contexts” [16]. A low co-occurrence frequency indicates a novel usage and thus, low relatedness between the mapped concepts. Other psychological features such as imageability [2, 3], meaningfulness [14] and familiarity [18] indicate the propensity of a word being metaphorical.

In this paper, we investigate the concept of metaphoricality for a given textual phrase through the notion of degree of membership in fuzzy clusters [1]. We contend that a soft metaphor detection approach based on metaphoricality would lead to a more informative metaphor processing system. We perform unsupervised soft metaphor detection which classifies sampled data into three classes namely, *metaphor*, *literal* and *probably_metaphor*. The ternary classification creates a subset of doubtful cases which requires further processing thus improving the confidence of classified instances. We use the R package *e1071* [9] to implement FCM for our experiments and use a publicly available dataset of nominal metaphors provided in [13] to analyze the validity of the proposed soft metaphor detection approach.

The remainder of the paper is organized as follows. Section 2 provides a brief introduction to existing work in metaphor detection. In Sect. 3, we explain the

¹ WordNet search (*gem*): <https://goo.gl/ej2PUY>.

application of FCM to determine metaphoricity for a textual input and report the results of a soft binary classification to identify metaphors in text. We conclude our work in Sect. 4.

2 Related Work

In this section, we discuss the prior studies on detection of nominal metaphors. Krishnakumaran et al. in [6] utilize WordNet [21] to verify the absence of hyponymy relations in the mapped subject and object to identify metaphorical usages. This establishes the variation in semantic categories and thus a non-literal mapping.

For example, consider the phrase ‘My lawyer is a shark’. In this phrase, the lawyer with semantic category ‘PERSON’ is projected as a type of shark which is an ‘ANIMAL’. Therefore, this phrase is marked as a metaphorical phrase.

However, the concept of hyponymy relation fails in case of polysemous words such as *chicken* which have multiple semantic categories *i.e.* ‘ANIMAL’ and ‘FOOD’. In addition, a phrase with same semantic category for the subject and object is not necessarily a literal phrase [11]. One such example is ‘My cat is a tiger.’ In [11], Neuman et al. resolved this problem by incorporating a disambiguation step which uses co-occurrence frequency to identify most likely usage. Su et al. combines the hyponymy relation from WordNet with cosine distance between the source and target domains using *word2vec* embeddings [17].

The utility of conceptual metaphors revolves around understanding of the abstract concept that is conveyed in the target domain by relating it with a relatively concrete concept in the source domain [7]. Employing this hypothesis, Turney et al. in [20] and Klebanov et al. in [5] use the notion of relative abstractness between a word and its context to detect metaphors. Other psychological features (also known as conceptual features) such as imageability [2, 14, 19] and familiarity [18] have been also shown to be helpful in identifying metaphors. Rai et al. use fuzzy psychological features and *word2vec* embeddings to identify nominal metaphors [15]. They later approximate the crisp classes namely metaphorical and literal using a fuzzy rough set model to tackle imprecision and vagueness in psychological features [13].

The existing studies on metaphor detection emphasize on contrasting the source and target domain concepts, without taking into account the degree of novelty of the metaphorical expression. To the best of our knowledge, there is no computational approach for any type of metaphors which explores the concept of fuzzy nature of a metaphor. In this paper, we introduce the concept of *metaphoricity* to represent the continuous spectrum of possibilities between the extremes, *literal* and *metaphor*. The approach closest to our work is that proposed in [4]. However, they adopt a probabilistic approach, quantifying the likelihood of a phrase being metaphorical. On the other hand, our approach assigns membership values that quantify the degree to which a phrase may belong to different, yet overlapping classes of metaphoricity. The fuzzy approach has

greater expressive power as it represents the realistic part-metaphorical, part-literal phases of a metaphorical expression in transition from its initial novel state to eventual dead state.

3 Soft Metaphor Detection Using Fuzzy c-Means

In this section, we elaborate upon our proposed FCM driven approach to model the concept of metaphoricity and enable an unsupervised soft classification of a textual input into the fuzzy sets *metaphor*, *literal* and *probably_metaphor*.

3.1 Problem Representation

In this paper, we restrict the problem of soft metaphor detection to nominal metaphors. In nominal metaphorical expressions, an explicit mapping is performed between the subject in the target domain and an object in the source domain. We use a dependency parser to extract the subject and object for all input utterances. The extracted pairs of <subject, object> are used for feature extraction in the next phase.

3.2 Feature Extraction

We extract a set of conceptual features namely *concreteness*, *imageability*, *familiarity* and *meaningfulness* for the source and target domains using MRC Psycholinguistic Database [22]. These are augmented by a set of derived features comprising the relative difference between the psychological features of the subject and the object to capture the extent of variation in the selected phrase.

We compute the cosine similarity between pre-trained *word2vec* embeddings [10] of the source and target domains to quantify the semantic relatedness that conveys the novelty of a mapping. We use word embeddings since they effectively capture context and analogical relations between the concepts.

3.3 Fuzzy c-Means

Let $X = \{x_1, x_2, x_3, \dots, x_N\}$ be a set of N sample points in an n -dimensional feature space \mathbb{R}^n . Let x_i^k represent the k^{th} feature of a sample point, x_i . Given a set of clusters $C = \{C_1, C_2, C_3, \dots, C_c\}$, the fuzzy c -partitions of X are such that each sample, $x_i \in X$ has a membership value, μ_{ij} in cluster, $c_j \in C$ with conditions such that,

$$\begin{aligned} \mu_{ij} &\in \{0, 1\}, \text{ where } 1 \leq i \leq N, 1 \leq j \leq c; \\ \sum_{j=1}^c \mu_{ij} &= 1, \text{ where } 1 \leq i \leq N; \\ \text{and } 0 < \sum_{i=1}^N \mu_{ij} &< N, \text{ where } 1 \leq j \leq c. \end{aligned}$$

The inclusion of $x_i \in X$ in a fuzzy partition, C_j is determined by minimizing the objective function defined in (1).

$$J_m(C, v) = \sum_{i=1}^N \sum_{j=1}^c (\mu_{ij})^m |x_i - v_j|^2. \tag{1}$$

where $v = \{v_1, v_2, \dots, v_c\}$ represents a vector for centre of clusters. m denotes weighting exponent to indicate the level of fuzziness, where $1 \leq m < \infty$. On the basis of number of clusters to be formed, the algorithm calculate the center, v_j of cluster, j , using (2).

$$v_j = \frac{\sum_{i=1}^N \mu_{ij}^m x_i}{\sum_{i=1}^N \mu_{ij}^m}. \tag{2}$$

Thereafter, randomly initialized membership, μ_{ij} of every sample point, $x_i \in X$ in every cluster $C_j \in C$ is updated using (3). The algorithm converges when the objective function, $J_m(C, v)$ ceases to improve by a fixed minimum threshold, $0 < \omega < 1$ or the specified number of iterations are over.

$$\mu_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{|x_i - v_j|}{|x_i - v_k|} \right)^{\frac{2}{m-1}}}. \tag{3}$$

In our case, there are two classes namely *metaphor* and *literal*, thus $c = 2$. The overlap between the clusters is considered to be *probably_metaphor* samples.

3.4 Experiments and Results

We used the R package *e1071* V1.6-8 [9] and *cluster* V2.0.6 [8] to implement FCM algorithm. We used a publicly available dataset of nominal metaphors provided in [13] for our experiments. In [13], the authors used the dataset to demonstrate a supervised metaphor detection model using fuzzy rough sets. Since, our approach is an unsupervised approach, we define a baseline to evaluate the effectiveness of our approach.

We show clusters formed using FCM clustering algorithm in Fig. 1. The cluster containing a majority of metaphorical samples is marked ‘1’ whereas the cluster containing mostly literal samples is marked ‘2’. For every sample in the dataset², we plotted their degree of membership in the created fuzzy clusters as shown in Fig. 2. The black line represents the membership value of instances for the cluster *literal* whereas the red line indicates the membership for the cluster *metaphor*.

For the baseline, we perform crisp classification using FCM. We mark a sample, i as *metaphor* if its membership values, μ_{M_i} in the cluster *metaphor* is higher than its membership, μ_{L_i} in the cluster *literal*. The results for soft metaphor detection are summarized in Table 1. We formed three classes namely *metaphor*, *literal* and *probably_metaphor*. The class, *metaphor* consisted of samples having

² Metaphoricity: <https://goo.gl/wmgjor>.

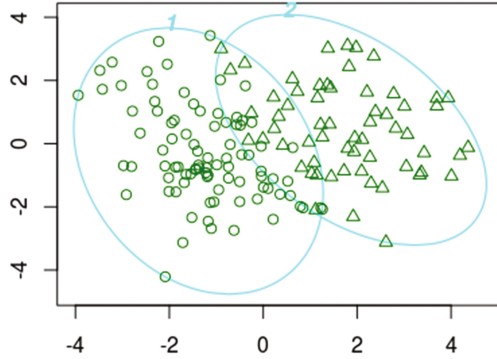


Fig. 1. Clustering using FCM (clusters: ‘1’ - Metaphor, ‘2’ - Literal)

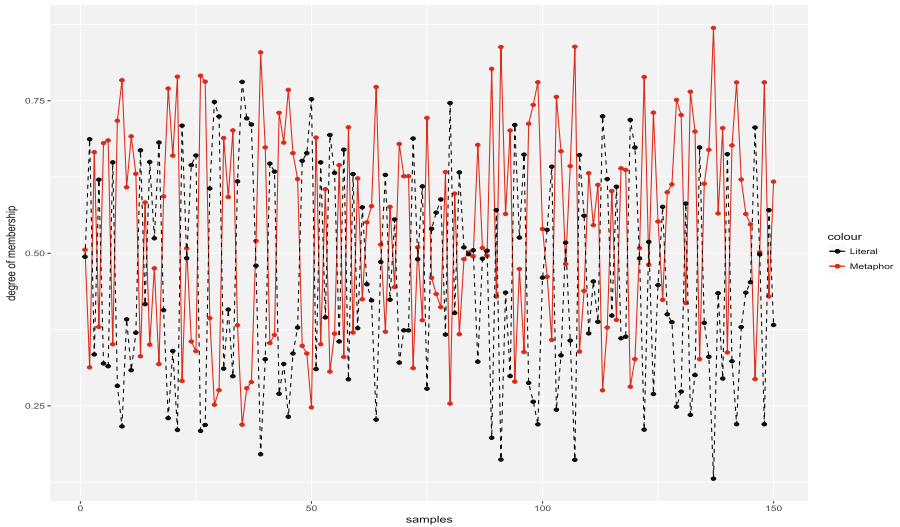


Fig. 2. Membership graph for metaphorical and literal samples. The x-axis represents the samples and y-axis indicates the degree of membership. (Color figure online)

membership value, ≥ 0.52 in cluster, *metaphor* where as samples with membership value ≤ 0.48 were marked as literal samples. The class, *probably_metaphor* comprised of samples whose membership values lie in the range of (0.48, 0.52). The range is decided experimentally after examining the samples.

In Table 1, the average accuracy for our proposed approach is 75.91% which is 3.91% more than the average accuracy of the baseline approach. We observe a significant improvement of 8.93% in the accuracy of class *metaphor* over the baseline. There is a minor increase of 0.8% in the accuracy of class *literal*.

Table 1. Results

Approach	$A_{metaphor}$	$A_{literal}$	$A_{average}$	$ probably_metaphor $
Baseline (crisp classification)	62.67	81.34	72	NA
Soft Metaphor Detection	71.6	82.14	75.91	13

Legend: $A_{metaphor}$ - Accuracy for class *metaphor*,

$A_{literal}$ - Accuracy for class *literal*,

$A_{average}$ - Average accuracy for *metaphor* and *literal* class,

$|probably_metaphor|$ - cardinality of set, *probably_metaphor*.

On closer analysis, we observe that metaphorical samples such as ‘The good news was an earthquake’ (sample: 48 in Fig. 2) and ‘An atom is a solar system’ (sample: 50 in Fig. 2) were incorrectly classified as *literal*. The degree of membership for the sample ‘50’ in cluster *literal* is 0.75 and thus 0.25 in *metaphor*. Despite the fact that it is a metaphorical sample, the frequent usage in daily parlance has nullified its novelty and thus its application as a metaphor. The sample ‘48’ has membership value of 0.65 in the cluster ‘literal’. This can be attributed to the high relatedness between the domains ‘news’ and ‘earthquake’. Also for a human, it is difficult to unambiguously categorize the sample ‘48’ as *metaphor* or *literal*.

Likewise, few literal samples such as ‘Marriage is a legal contract.’ are predicted as *metaphor*. Technically, marriage is a contract but it is seldom conveyed so. Low co-occurrence and thereby low relatedness between *marriage* and *contract* led to its false classification in the class *metaphor*.

From the results, we observe that soft metaphor detection do facilitate classification with higher accuracy and provides the scope for further analytical processing of doubtful cases.

The degree of membership in the fuzzy cluster ‘1’ (*metaphor*) acts as the *metaphoricity* of a given metaphorical expression. In Table 2, we present the metaphoricity obtained for a subset of 6 samples from the dataset. The table includes two pairs of similar subject-object phrases, one embodying a much higher degree of metaphoricity than the other. These examples illustrate how an unfamiliarity in the mapping between source and target domain concepts bends a phrase towards metaphorical usage. The complete list is publicly available on the link (see Footnote 2).

In Table 3, we show the centroids of clusters in terms of the feature-values. The labels *conc*, *imag*, *mean* and *fam* denote the features *concreteness*, *imageability*, *meaningfulness* and *familiarity* respectively. The suffix *_o* indicates that the extracted feature is for target domain *i.e.* object in the case of nominal metaphors whereas the suffix *_d* indicates the difference between the values for the source and target domains.

Analysis. Analyzing the clusters shown in Fig. 1, we observe that there is a significant overlap between the metaphor and literal clusters. The membership graph shown in Fig. 2 also strengthens the idea that a metaphorical text is

Table 2. Metaphoricity (membership in clusters-*metaphor*)

No. (Fig. 2)	Sample	Metaphoricity
9	New moon is a <i>banana</i>	0.784
109	New moon is a curve	0.438
31	His marriage was a short <i>leash</i>	0.689
94	His marriage was controlling	0.29
47	Control is <i>fertilizer</i>	0.622
96	Control is encouraging	0.338

Table 3. Feature-value for cluster-centroids in FCM

Feature	<i>conc_o</i>	<i>imag_o</i>	<i>mean_o</i>	<i>fam_o</i>	<i>conc_d</i>	<i>imag_d</i>	<i>relatedness</i>
Metaphor	0.772	0.698	0.502	0.656	0.633	0.498	0.177
Literal	0.538	0.447	0.277	0.604	0.498	0.371	0.183

indeed partially literal and a literal text is partially metaphorical. The highest metaphoricity is 0.869 for ‘*Path through forest is a narrow lane.*’ followed by the phrase ‘*The nearest star is a ball.*’. The lowest metaphoricity is 0.219 for ‘*Some tears are intriguing.*’ followed by ‘*Atom is a solar system.*’. Few examples of *probably_metaphor* are given below.

- My rat’s fur is silk. (b)
 Brain is a machine (c)
 Some snores are sirens. (d)

The sentences (b–d) are marked *metaphor* in the dataset. However, the metaphoricity for these three phrases are 0.482, 0.495 and 0.501 respectively. This supports our hypothesis that a metaphor gradually loses its novelty and thereby, its metaphoricity. The comparisons *atom-solar system*, *fur-silk* and *brain-machine* are quite common and so have low metaphoricity. On the other hand, we also observe that some literal samples such as ‘*My young cousin is thin.*’, ‘*My ex-husband is good.*’ and ‘*Hostility is hidden.*’ have higher membership in the metaphor class. This may be due to the assumption of rare co-occurrences as an indicator of novelty and thus a metaphorical text.

As shown in Table 3, the psychological features such as *concreteness*, *imageability*, *familiarity*, *meaningfulness* are relatively high for metaphorical class in concordance with the existing findings. The semantic relatedness for metaphorical samples is relatively lower than literal samples, as postulated by the theory of contextual incongruity. The relative difference between psychological features of source and target domains is also an important feature to determine metaphorical samples.

4 Conclusion

In this paper, we brought forth the idea that the journey of a metaphor from being novel to being considered dead is a continuum. We argued that soft metaphor classification, which entails assigning a degree of membership to various levels of metaphoricity is a more practical and informative approach towards metaphor processing than a crisp classification, due to the fuzzy nature of concepts in human language. Through an analysis of cluster formation, we verified the hypothesis that metaphors do involve an analogous comparison of concepts in a somewhat inscrutable domain with concepts in a relatively more concrete, imageable and meaningful domain. For our future work, we are focusing on treating the intermediate category of *probably_metaphor* to further analytical methods to predict their utility as metaphorical usage.

References

1. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms, Ed. 1, Springer US (1981). <https://doi.org/10.1007/978-1-4757-0450-1>
2. Bracewell, D.B., Tomlinson, M.T., Mohler, M., Rink, B.: A tiered approach to the recognition of metaphor. In: CICLing, vol. 1, pp. 403–414 (2014)
3. Broadwell, G.A., et al.: Using imageability and topic chaining to locate metaphors in linguistic corpora. In: Greenberg, A.M., Kennedy, W.G., Bos, N.D. (eds.) SBP 2013. LNCS, vol. 7812, pp. 102–110. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37210-0_12
4. Dunn, J.: Measuring metaphoricity. In: ACL, vol. 2, pp. 745–751 (2014)
5. Klebanov, B.B., Leong, C.W., Flor, M.: Supervised word-level metaphor detection: experiments with concreteness and reweighting of examples. In: Proceedings of the Third Workshop on Metaphor in NLP, pp. 11–20 (2015)
6. Krishnakumaran, S., Zhu, X.: Hunting elusive metaphors using lexical resources. In: Proceedings of the Workshop on Computational Approaches to Figurative Language, pp. 13–20. Association for Computational Linguistics (2007)
7. Lakoff, G., Johnson, M.: Metaphors We Live By. University of Chicago Press, Chicago (2008)
8. Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K.: Cluster: cluster analysis basics and extensions. R package version 2.0.1.2015 (2017)
9. Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C.C., Lin, C.C., Meyer, M.D.: Package ‘e1071’ (2017)
10. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
11. Neuman, Y., Assaf, D., Cohen, Y., Last, M., Argamon, S., Howard, N., Frieder, O.: Metaphor identification in large texts corpora. PLoS One **8**(4), e62343 (2013)
12. Nunberg, G.: Poetic and prosaic metaphors. In: Proceedings of the 1987 Workshop on Theoretical Issues in Natural Language Processing, pp. 198–201. Association for Computational Linguistics (1987)
13. Rai, S., Chakraverty, S.: Metaphor detection using fuzzy rough sets. In: Polkowski, L., Yao, Y., Artiemjew, P., Ciucci, D., Liu, D., Ślęzak, D., Zielosko, B. (eds.) IJCRS 2017. LNCS (LNAI), vol. 10313, pp. 271–279. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60837-2_23

14. Rai, S., Chakraverty, S., Tayal, D.K.: Supervised metaphor detection using conditional random fields. In: Proceedings of the Fourth Workshop on Metaphor in NLP, pp. 18–27. Association of Computational Linguistics (2016)
15. Rai, S., Chakraverty, S., Tayal, D.K.: Identifying metaphors using fuzzy conceptual features. In: Kaushik, S., Gupta, D., Kharb, L., Chahal, D. (eds.) ICICCT 2017. ICICCT 2017, vol. 750, pp. 379–386. Springer, Heidelberg (2017). https://doi.org/10.1007/978-981-10-6544-6_34
16. Rubenstein, H., Goodenough, J.B.: Contextual correlates of synonymy. *Commun. ACM* **8**(10), 627–633 (1965)
17. Su, C., Huang, S., Chen, Y.: Automatic detection and interpretation of nominal metaphor based on the theory of meaning. *Neurocomputing* **219**, 300–311 (2017)
18. Thibodeau, P.H., Durgin, F.H.: Metaphor aptness and conventionality: a processing fluency account. *Metaphor Symbol* **26**(3), 206–226 (2011)
19. Tsvetkov, Y., Boytsov, L., Gershman, A., Nyberg, E., Dyer, C.: Metaphor detection with cross-lingual model transfer, In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 248–258, Baltimore, Maryland, USA, 23–25 June 2014. Association for Computational Linguistics.
20. Turney, P.D., Neuman, Y., Assaf, D., Cohen, Y.: Literal and metaphorical sense identification through concrete and abstract context. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 680–690. Association for Computational Linguistics (2011)
21. Princeton University: About WordNet (2010). <http://wordnet.princeton.edu>
22. Wilson, M.: MRC psycholinguistic database: machine-usable dictionary, version 2.00. *Behav. Res. Meth.* **20**(1), 6–10 (1988)

A Study on CART Based on Maximum Probabilistic-Based Rough Set

Utpal Pal^(✉), Sharmistha Bhattacharya (Halder), and Kalyani Debnath

Tripura University, Tripura, India

{utpalpal,s.bhattacharya}@tripurauniv.in, dkalyanimath@gmail.com

Abstract. The Classification and Regression Tree (CART) recursively partitions the measurement space, displaying the resulting partitions as decision tree. However, the performance of CART-based decision tree degrades while dealing with high-dimensional large data sets. This research work studies CART, based on Maximum Probabilistic-based Rough Set (MPBRs). The MPBRs has been used as a tool for insignificant data reduction without sacrificing information content. This paper also studies CART, based on Pawlak rough set and Bayesian Decision Theoretic Rough Set (BDTRS) for comparative analysis. Experimental results on three different data sets show that the MPBRs-based CART constructs improved decision tree for better classification efficiency.

Keywords: CART · MPBRs · BDTRS · Attribute reduction

1 Introduction

The CART [3, 10, 17] is a sophisticated Decision Tree [11, 12] algorithm. It is direct viewing, can extract knowledge rules preferably and suitable to deal with classification problems [21]. It outperforms several other popular decision tree algorithms with respect to simplicity, comprehensibility, classification accuracy and being able to handle mixed-type data. Moreover CART efficiently deals with missing values, uses cost-complexity pruning strategy and can handle outliers. But, when the data has large number of attributes and involves impurity, the decision tree constitutive property is poor and difficult to find some information that could have been found and be useful. Another problem with the CART cross validation method is that it can be computationally too expensive, because it requires the growing and pruning of auxiliary trees as well. In order to overcome these drawbacks, rough set based attribute reduction has been introduced. Ever since the introduction of rough set theory by Pawlak [9] in 1982, many extensions have been made [16, 19, 20, 22]. The MPBRs [7, 9] is an excellent rough set based tool for attribute reduction. The indiscernibility relation and set approximation remains unaltered before and after reduction. The positive region computed, involves maximum number of objects. As a result the reduced attribute set involves all the significant attributes eliminating all insignificant attributes. This paper also studies CART based on Pawlak rough set [9] and BDTRS [1, 2, 8] for attribute reduction. In this research work we have used R [4, 6, 13], version 3.2.2, for experimentation and implementation of MPBRs. Data sets have been taken from UCI Machine learning repository.

In Sect. 2, Pawlak rough set, BDTRS, MPBRS and CART are introduced. Section 3, discusses about processing steps, algorithm and execution process in R environment. Experimental results and concluding remarks are presented in Sects. 4 and 5 respectively.

2 Theoretical Background

2.1 Pawlak’s Rough Set Model [9, 13]

An information system is defined as: $S = (U, At, \{V_a | a \in At\}, \{I_a | a \in At\})$, where, U is a finite nonempty set of objects, At is a finite nonempty set of attributes, V_a is a nonempty set values of $a \in At$ and $I_a: U \rightarrow V_a$ is an information function that maps an object in U to exactly one value in V_a . The approximations of $X \subseteq U$ with respect of equivalence relation R can be defined according to its upper and lower approximations.

Positive region:

$$POS_R(X) = \underline{R}X = \cup \{[x]_R | P(X/[x]_R) = 1, [x]_R \in \pi_R\}.$$

Negative region:

$$NEG_R(X) = U - \bar{R}X = \cup \{[x]_R | P(X/[x]_R) = 0, [x]_R \in \pi_R\}.$$

Boundary region:

$$BND_R(X) = \bar{R}X - \underline{R}X = \cup \{[x]_R | 0 < P(X/[x]_R) < 1, [x]_R \in \pi_R\}.$$

2.2 Bayesian Decision Theoretic Rough Set [1, 2, 8, 15]

Let, D_{POS} denotes the positive region in BDTRS model. For an equivalence class, $[x]_c \in \pi_A$,

$$D_{POS}([x]_c) = \{D_i \in \pi_D : P(D_i/[x]_c) > P(D_i)\}.$$

For equivalence classes $[x]_c$ and $[y]_c$ the elements of a positive decision-based discernibility matrix, $M_{D_{pos}}$ is defined as follows.

$$M_{D_{pos}}([x]_c, [y]_c) = \left\{ a \in C : I_a(x) \neq I_a(y) \wedge D_{POS}([x]_c) \neq D_{POS}([y]_c) \right\}.$$

A positive decision reduct is a prime implicant of the reduced disjunctive form of the discernibility function.

$$f(M_{D_{pos}}) = \bigwedge \left\{ \bigvee (M_{D_{pos}}([x]_c, [y]_c)) : \forall x, y \in U (M_{D_{pos}}([x]_c, [y]_c) \neq \emptyset) \right\}.$$

In order to derive the reduced disjunctive form, the discernibility function $f(M_{D_{pos}})$ is transformed by using the absorption and distributive laws. Accordingly, finding the set of reducts can be modeled based on the manipulation of a Boolean function.

2.3 Maximum Probabilistic Based Rough Set [7, 9, 18]

Maximum probabilistic based rough set is a stronger form of other rough set models.

The precision of an equivalence class $[x]_c \in \pi_c$ for predicting a decision class $D_i \in \pi_D$ can be defined. We denote it by $P_{max}(D_i, [x]_c)$ and are defined as follows:

$$P_{max}(D_i, [x]_c) = \frac{|[x]_c \cap D_i|}{\max_j |[x]_c \cap D_i|}.$$

For a decision class, $D_i \in \pi_D$, the Maximum probabilistic based rough set lower and upper approximations with respect to a partition π_c can be defined as:

$$\begin{aligned} \underline{apr}_{\max(\pi_c)}(D_i) &= \left\{ x \in U : P_{max}(D_i, [x]_c) = \frac{|[x]_c \cap D_i|}{\max_j |[x]_c \cap D_i|} = 1 \right\}, \\ \overline{apr}_{\max(\pi_c)}(D_i) &= \left\{ x \in U : P_{max}(D_i, [x]_c) = \frac{|[x]_c \cap D_i|}{\max_j |[x]_c \cap D_i|} > 0 \right\}. \end{aligned}$$

The positive, boundary and negative regions of $D_i \in \pi_D$ with respect to π_c are defined by:

$$\begin{aligned}
 POS_{max(\pi_c)}(D_i) &= POS_{max}(D_i, \pi_c) = \left\{ x \in U : P_{max}(D_i, [x]_c) = \frac{|[x]_c \cap D_i|}{\max_j |[x]_c \cap D_i|} = 1 \right\} \\
 BND_{max(\pi_c)}(D_i) &= BND_{max}(D_i, \pi_c) \\
 &= \left\{ x \in U : 0 < P_{max}(D_i, [x]_c) = \frac{|[x]_c \cap D_i|}{\max_j |[x]_c \cap D_i|} < 1 \right\} \\
 NEG_{max(\pi_c)}(D_i) &= NEG_{max}(D_i, \pi_c) = \left\{ x \in U : P_{max}(D_i, [x]_c) = \frac{|[x]_c \cap D_i|}{\max_j |[x]_c \cap D_i|} = 0 \right\}.
 \end{aligned}$$

Attribute Significance [9]. The consistency factor is defined as $\gamma(C, D) = |POS_C(D)|/|U|$. The decision table is consistent if $\gamma(C, D) = 1$. The significance, $\sigma(a)$, of any attributes a , can be defined as:

$$\sigma(C, D)(a) = (\gamma(C, D) - \gamma(C - \{a\}, D))/\gamma(C, D) = 1 - (\gamma(C - \{a\}, D)/\gamma(C, D)).$$

Where, $0 \leq \sigma(a) \leq 1$.

2.4 Decision Tree: CART [3, 10, 11, 17]

The CART algorithm starts with the initial decision table (data set) D , attribute set A and *gini-index*, as attribute selection method. Initially, it creates a node, N , which incorporates D . If all the objects of D belong to same class, node N is returned as leaf node. Otherwise, select a condition attribute (that maximizes the reduction in impurity of D) that divides D , in a manner such that height of the tree is as small as possible. The node N is labeled with the selected attribute. After that, branches are grown from N for each of the outcomes of the splitting attributes. The algorithm works recursively on each subset of D . Recursion may stop in one of these cases:

- All the objects of a node belong to a particular class (same class).
- There are no attributes left on which the objects may be further be separated.
- There are no objects for a particular branch and a partition is empty.

Gini Index. CART uses *Gini-index* for choosing the best attribute in the process of data partition. The *Gini-index* for a data set D , having m decision classes is calculated as:

$$Gini(D) = 1 - \sum_{i=1}^m p_i^2. \quad (1)$$

Where p_i is the probability that an object in D , belongs to class C . If a binary partition on condition attribute A , partitions D into subsets D_1 and D_2 , the *Gini-index*, given that partitioning is:

$$Gini_A(D) = \frac{|D_1|}{D} Gini(D_1) + \frac{|D_2|}{D} Gini(D_2). \quad (2)$$

For every attribute, all of the possible splits are calculated. For a particular attribute, the point producing the lowest *Gini index* is chosen as the split point. Reduction in impurity on a condition attribute A , is:

$$\Delta Gini(A) = Gini(D) - Gini_A(D). \quad (3)$$

The condition attribute that results in maximum reduction of impurity is chosen as the splitting attribute.

3 Implementation of MPBRS-Based CART

3.1 Processing Steps

This section shows the implementation and execution procedure for MPBRS-based CART. Implementation of BDTRS is done in [8]. MPBRS-based decision tree induction is performed in two basic steps. First, attribute reduction and second, decision tree induction using the reduced information. Procedure for attribute reduction is shown in Algorithm-1(AttReduction ()). Case: 1, Case: 2 and Case: 3 represent MPBRS, BDTRS and Pawlak rough set respectively for attribute reduction. Based on rough set theory equivalent classes [9] are computed. Procedure for computation of positive region, discernibility matrix [15], discernibility function [19] and reduced attribute set are shown in Algorithm-1. The discernibility function is a conjunction over the disjunction of the matrix elements. The function can be transformed in to a reduced attribute set using absorption and distributive laws of Boolean algebra. The classical CART is implemented in package ‘rpart’ of R. Induction of decision tree using R commands has been shown in Sect. 3.3.

3.2 Algorithm

Algorithm-1. AttReduction (DT):

Input: Decision Table (DT), Decision column (D), Data Matrix (D_m).

Output: Reduced data set, D_r .

Variables D_{pos} , represents Positive Region. X , $objIdx$, $objIdx1$, $objIdx2$ represents intermediate variables.

1. Compute $[IND]$, the Indiscernible relation/Equivalent classes from DT .
2. Repeat step 3 for $[IND_i]$, $i=1$ to n , n being the number of Equivalent classes.
3. Repeat 3.1 to 3.2 for each decision (D_j), $j=1$ to k , k is number of unique decision.
 - Case: 1 [For Maximum Probabilistic-based Rough Set]
 - 3.1. $X = (\text{Objects in } [IND_i] \text{ having } D_j) \div \text{Max}(\text{Objects in } [IND_i] \text{ having } D_j)$
 - 3.2. If ($X = 1$)
 - Assign Decision D_j , to D_{pos} , corresponding to each object of $[IND_i]$
 - Case: 2 [For Bayesian Decision Theoretic Rough Set]
 - 3.1. $X = (\text{Objects in } [IND_i] \text{ having } D_j \text{ as decision} \div \text{Objects in } [IND_i])$
 - 3.2. If ($X > P(D_j)$) [$P(D_j)$ is the probability of D_{jth} Decision]
 - Assign Decision D_j , to D_{pos} , corresponding to each object of $[IND_i]$
 - Case: 3 [For Pawlak Rough Set]
 - 3.1. $X = (\text{Objects in } [IND_i] \text{ having } D_j \text{ as decision} \div \text{Objects in } [IND_i])$
 - 3.2. If ($X = 1$)
 - Assign Decision D_j , to D_{pos} , corresponding to each object of $[IND_i]$.
4. Repeat for i^{th} object in DT , where $i = 1$ to $(n-1)$, n is the number of objects in DT
 - 4.1. Find the objects (in DT) whose decision in D doesn't match with the decision of i^{th} object. [i.e. $objIdx1 = (\text{decVector}[i] \neq (\text{decVector}[i+1] \text{ to } \text{decVector}[n]))$]
 - 4.2. Find objects (in DT) whose decision in D_{pos} doesn't match with the decision of i^{th} object.
 - [i.e. $objIdx2 = (D_{pos}[i] \neq (D_{pos}[i+1] \text{ to } D_{pos}[n]))$]
 - 4.3. $objIdx = \{ objIdx1 \cap objIdx2 \}$ [Common objects].
 - 4.4. Repeat for j^{th} object in $objIdx$, where $j = 1$ to no. of objects in $objIdx$
 - 4.4.1 $A_l = \text{Column names of } D_m \text{ where Column name of } i^{th} \text{ object} \neq \text{Column name of } j^{th} \text{ object.}$
 - 4.4.2 $D_{mat} = A_l$ [Assign the attribute list A_l to Discernibility Matrix D_{mat}].
5. Compute discernibility function, D_f from D_{mat}
6. Transform D_f into reduced attribute set, A_r (using laws of Boolean Algebra)
7. $D_r = \text{Data set corresponding to the attributes of } A_r$.
8. Return (D_r).

3.3 Execution of CART and MPBRS-Based CART in R Environment

Decision Tree Induction Using CART. This section explains decision tree induction by taking *housing* [5] as example data set. Installation procedure of R and relevant packages (“RoughSets” [14], “rpart” etc.) are available in Comprehensive R Archive Network (CRAN). In order to perform attribute reduction, raw data (in .txt, .csv, .xlsx etc. format) is first converted into *data frame* object which is then converted into *DecisionTable* format. For this, functions like *read.table()*, *SF.asDecisionTable()* [14] may be used. The package “rpart” is installed using the R command: `> library(rpart)`. Similarly other packages like “rattle”, “caret”, “RoughSets” etc. are also installed. The Decision Tree, T_1 obtained from the following commands is shown in Fig. 1.

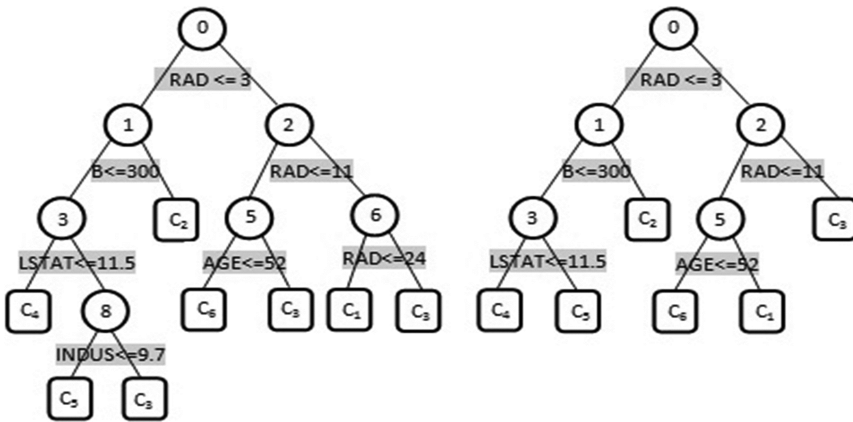


Fig. 1. Decision Tree T_1 using CART method (Left hand side), before attribute reduction and Decision Tree T_2 using MPBRS-CART() method (Right hand side) after attribute reduction.

```
>HData = RoughSetData$housing.dt#Using housing DataSet
>train.flag = createDataPartition(y = HData
  $MEDV,p=0.7,list=FALSE) #Training and test sample
>training = HData [train.flag,]
>Validation = HData [-train.flag,]
>modfit = train(MEDV~.,method="rpart",data=training)
#Building the Mmodel
>fancyRpartPlot(modfit$finalModel)#DrawCART Decision Tree
```

Decision Tree Induction Using MPBRS-Based CART. MPBRS-based CART is induced in two steps. Step 1: Performs data reduction by MPBRS. Step2: Decision Tree induction based on the output of Step 1. Following R commands are executed on *housing* data set (*housing.csv* format). Reduced attribute set is shown in Table 1.

Table 1. Attribute reduction using MPBRS model on *housing* dataset.

Attribute set of D before reduction	Total	Attribute set of D_{red} after reduction	Total
{RAD, RM, DIS, B, AGE, CRIM, CHAS, LSTAT, TAX, ZN, NOX, PTRATIO, INDUS}	13	{RAD, RM, DIS, B, AGE, CRIM, CHAS, LSTAT, TAX}	09

Computed attribute significance in ascending order: B: 0.75, RAD: 0.74, RM: 0.74, LSTAT: 0.70, AGE: 0.68, CHAS: 0.50, CRIM: 0.50, TAX: 0.49 DIS: 0.42, ZN: 0.12, NOX: 0.08, PTRATIO: 0.06, INDUS: 0.02.

```
>HousingFrm = read.table ("housing.csv", header = TRUE,
sep = ",") #Computation of DataFrame
>HousingDecTable = SF.asDecisionTable (HousingFrm,
decision.attr = 14, indx.nominal = c(1:13) #to Decision
table
>MPBRS_PosRegion = BayesDtPos (HousingDecTable, c(1:13))
#Computation of Positive Region
>MPBRS_DisMatrix = BayesDtDisMat (HousingDecTable,
MPBRS_PosRegion, range.object = NULL, return.matrix
=TRUE)# Computation of Discernibility Matrix
>ReducedDataSet=FS.one.reduct.computation(MPBRS_DisMatrix)
#Reduced attribute set
```

The procedure for decision tree induction is same as CART and hence not repeated. The decision tree (T_2) thus obtained is shown in Fig. 1.

4 Experimental Results and Discussion

The data sets used for experimentation are: *Cervical Cancer* (858 objects with 36 attributes) [23], *Spambase* (4601 objects with 57 attributes) [24] and *housing* (506 objects with 14 attributes). The housing data set is already introduced in the previous section. Each of these data sets has been studied by MPBRS, BDTRS and Pawlak rough set. For housing data, at first, we pre-processed the sample data (D) and filled in missing values using built in functions available in R. Next, we run the original algorithm CART, on D , to construct a Decision Tree T_1 as shown in Fig. 1.

After that, we run Case: 1 of Algorithm-1 (attribute reduction by MPBRS) to reduce the insignificant attributes of D . We reduced the number of attributes down to 9, saved the new sample data set as D_{red} . The deleted attributes are: 'ZN', 'NOX', 'PTRATIO' and 'INDUS'. We computed attribute significance of all the attributes to show that deleted attributes have little effect on decision making. This is shown in Table 1. We further computed consistency factor (C.F) (shown in Table 2) which remains same (one) before and after attribute reduction. This ensures that the integrity of the data set remains unchanged after attribute reduction. Finally, we run CART on the reduced data set D_{red} , to construct a simplified decision tree, T_2 (Fig. 1). The above mentioned experimental procedure is repeated using the functions AttReduction() (Algorithm-1) and CART

method for *Cervical Cancer* and *Spambase* data sets. The results obtained from the computations are shown in Table 2.

Table 2. Comparison of CART, MPBRS-CART, BDTRS-CART and Pawlak-CART using *Cervical Cancer*, *Spambase* and *housing* data sets.

Data set used	Model used	Number of condition attributes	Depth of tree	Number of leaves	Average length of rules	Average Exe. time in seconds	Classification accuracy	C.F
<i>Cervical Cancer</i>	CART	36	10	09	8.24	0.98	90.23%	1
	MPBRS-CART	26	09	08	8.01	1.32	91.89%	1
	BDTRS-CART	27	09	08	8.03	1.28	91.89%	1
	Pawlak-CART	32	10	09	8.22	1.12	91.05%	1
<i>Spam- base</i>	CART	57	21	20	14.33	2.6	86.45%	1
	MPBRS-CART	44	18	17	13.12	3.12	88.12%	1
	BDTRS-CART	49	19	18	13.38	3.12	88.02%	1
	Pawlak-CART	52	20	19	14.01	3.03	87.10%	1
<i>housing</i>	CART	13	05	04	3.12	0.59	88.65%	1
	MPBRS-CART	09	04	03	2.88	0.92	88.75%	1
	BDTRS-CART	09	04	03	2.88	0.92	88.70%	1
	Pawlak-CART	12	05	04	2.98	0.88	88.70%	1

For *housing* data, tree T_1 , shows that the classification tree has 15 nodes (7 internal and 8 leaf nodes). On the other hand, Tree T_2 , of Fig. 1 shows that the classification tree has 11 nodes (5 internal and 6 leaf nodes). The domain of decision column (Minimum: 5 and Maximum: 50) is divided in to six equivalent classes: C_1, C_2, C_3, C_4, C_5 and C_6 . There are eight (8) classification rules corresponding to the leaf nodes of T_1 and six (6) rules corresponds to T_2 . For example, the classification rule: “If (RAD \leq 3) and (B \leq 300) and (LSTAT $>$ 11.5) and (INDUS \leq 9.7) then MEDV = C_5 ”, corresponds to leaf node C_5 of tree T_1 . Similar results obtained from the other two data sets and the comparison of CART, MPBRS-CART, BDTRS-CART and Pawlak-CART methods are shown in the Table 2. It can be observed from Table 2 that CART model deals with the original unreduced data set, whereas, other three approaches work on the reduced data set. The MPBRS method gives the best attribute reduction. As a result, number of nodes, number of leaf nodes, depth and average length of the classification rules has been decreased for all the three approaches except classical CART Model. This ultimately improves the classification accuracy of the decision trees as shown in Fig. 2.

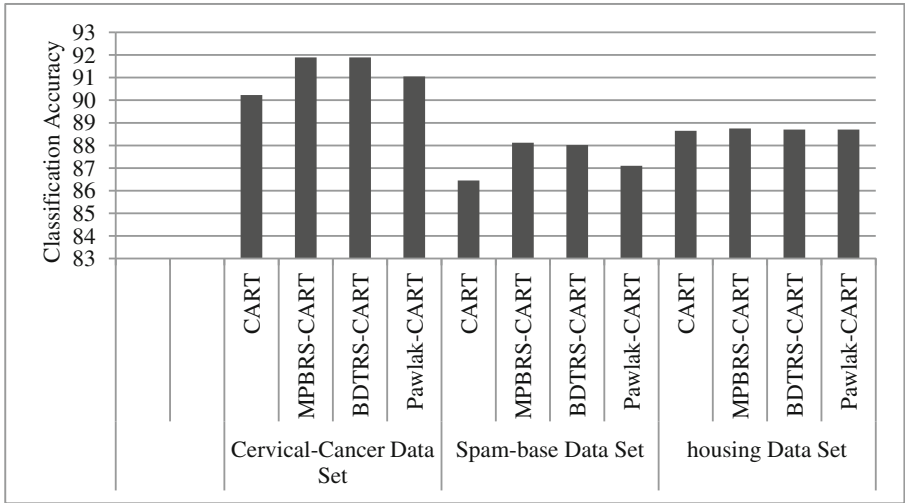


Fig. 2. Classification accuracy of CART, MPBRS-CART, BDTRS-CART and Pawlak based CART on *Cervical Cancer*, *Spambase* and *housing* data sets.

On the other hand, the rough set based decision tree approach suffers in terms of total execution time as it involves the attribute reduction phase also. The minimal increase of execution time is acceptable, keeping the classification accuracy and reduction of tree complexity (shown in Fig. 3) in mind. Tree complexity mainly depends on the average length of the decision rules which is lowest in case of MPBRS-based CART method for all data sets.

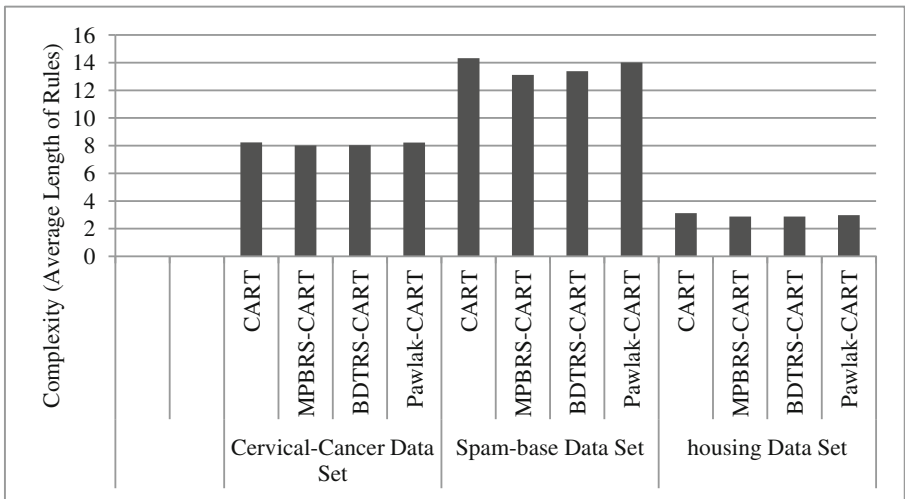


Fig. 3. Complexity representation of CART, MPBRS-CART, BDTRS-CART and Pawlak based CART on *Cervical Cancer*, *Spambase* and *housing* data sets.

5 Conclusion

Efficiency of CART-based decision tree becomes an issue of concern to deal with high-dimensional large data sets. This study focuses on reducing number of insignificant attributes from the original data set before induction of CART-based decision tree. The reduced attribute set preserves the indiscernibility relation and set approximation. This is ensured by computing attribute significance and consistency factor. In this research work we have also implemented the MPBRS using R language in order to study the classical CART. The experimental results show that the decision tree induced by MPBRS-based CART is the simplest and most efficient in terms of depth, number of nodes, average rule length and classification accuracy compared to the other methods mentioned in this work.

References

1. Bhattacharya (Halder), S.: A study on Bayesian decision theoretic rough set. *Int. J. Rough Sets Data Anal. (IJRSDA)* **1**(1), 1–14 (2014)
2. Bhattacharya (Halder), S., Debnath, K.: Attribute reduction using Bayesian decision theoretic rough set models. *Int. J. Rough Sets Data Anal. (IJRSDA)* **1**(1), 15–31 (2014)
3. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: *Classification and Regression Trees*. Wadsworth & Brooks/Cole Advanced Books & Software, Monterey (1984). ISBN 978-0-412-04841-8
4. Ferraro, M.B., Giordani, P.: A toolbox for fuzzy clustering using the R programming language. *Fuzzy Sets Syst.* **279**, 1–16 (2015). Elsevier
5. Harrison, D., Rubinfeld, D.L.: Hedonic prices and the demand for clean air. *J. Environ. Econ. Manag.* **5**, 81–102 (1978)
6. Ihaka, R., Gentleman, R.: R: a language for data analysis and graphics. *J. Comput. Graph. Stat.* **5**, 299–314 (1996)
7. Pal, U., Bhattacharya (Halder), S., Debnath, K.: A Study on Maximum Probabilistic Based Rough Set (MPBRS), Communicated
8. Pal, U., Bhattacharya (Halder), S., Debnath, K.: R implementation of bayesian decision theoretic rough set model for attribute reduction. In: Bhattacharyya, S., Sen, S., Dutta, M., Biswas, P., Chattopadhyay, H. (eds.) *Industry Interactive Innovations in Science, Engineering and Technology*. LNNS, vol. 11, pp. 459–466. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-3953-9_44
9. Pawlak, Z.: Rough sets. *Int. J. Comput. Inform. Sci.* **11**, 341–356 (1982)
10. Questier, F., Put, R., Coomans, D., Walczak, B., Vander Heyden, Y.: The use of CART and multivariate regression trees for supervised and unsupervised feature selection. *Chemometr. Intell. Lab. Syst.* **76**(1), 45–54 (2005). <https://doi.org/10.1016/j.chemolab.2004.09.003>. ISSN 0169-7439
11. Quinlan, J.R.: Induction of decision trees. *Mach. Learn.* **1**, 81–106 (1986)
12. Quinlan, J.R.: Simplifying decision trees. *Int. J. Man-Mach. Stud.* **27**, 221–234 (1987)
13. R Development Core Team: *R: A Language and Environment for Statistical Computing*. The R Foundation for Statistical Computing, Vienna (2011). <http://www.R-project.org/>. Accessed 08 June 2016. ISBN 3-900051-07-0

14. Riza, L.S., Janusz, A., Bergmeir, C., Cornelis, C., Herrera, F., Slezak, D., Benitez, J.M.: Implementing algorithms of rough set theory and fuzzy rough set theory in the R package “RoughSets”. *Inf. Sci.* **287**, 68–89 (2014). Elsevier
15. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Slowiski, R. (ed.) *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory*, pp. 311–362. Kluwer Academic Publishers, Dordrecht (1992)
16. Slezak, D., Ziarko, W.: Bayesian rough set model. In: *Proceedings of the International Workshop on Foundation of Data mining, Japan*, pp. 131–135 (2002)
17. Stuart, L.C.: Extensions to the CART algorithm. *Int. J. Man-Mach. Stud.* **31**(2), 197–217 (1989). [https://doi.org/10.1016/0020-7373\(89\)](https://doi.org/10.1016/0020-7373(89)). ISSN 0020-7373
18. Yao, Y.Y.: Generalized rough set models. In: Polkowski, L., Skowron, A. (eds.) *Rough Sets in Knowledge Discovery*, pp. 286–318. Physica-Verlag, Heidelberg (1998)
19. Yao, Y.Y.: Probabilistic approaches on rough sets. *Expert Syst.* **20**, 287–297 (2003)
20. Yao, Y.Y., Wong, S.K., Lingras, P.: A decision theoretic rough set model. In: Ras, Z.W., Zemankova, M., Emrich, M.L. (eds.) *Methodologies for Intelligent Systems*, vol. 5, pp. 17–24. North Holland, New York (1990)
21. Zhiling, C., Qingmin, Z., Qinglian, Y.: A method based on rough set to construct decision tree. *J. Nanjing Univ. Technol.* **27**, 80–83 (2005)
22. Ziarko, W.: Variable precision rough set model. *J. Comput. Syst. Sci.* **46**, 39–59 (1993)
23. <https://archive.ics.uci.edu/ml/datasets/Cervical+cancer+%28Risk+Factors%29>. Accessed 21 June 2017
24. <http://archive.ics.uci.edu/ml/datasets/Spambase>. Accessed 21 June 2017

Portfolio Optimization in Dynamic Environments Using MemSPEAI

Priyank Shah^(✉) and Sanket Shah

Dwarkadas J. Sanghvi College of Engineering, University of Mumbai, Mumbai, India
priyank3112@gmail.com, sanketshah33@yahoo.in

Abstract. This paper concerns the problem of portfolio optimization in dynamic environments using multi-objective evolutionary algorithms. Financial markets are characterized by volatility and uncertainty making portfolio optimization a challenging task. A novel multi-objective genetic programming algorithm is proposed, which is a memory enhanced version of the standard SPEAI algorithm. The proposed algorithm employs an explicit memory to store a number of non-dominated solutions. These solutions are reused in the later stages for adapting to the changing environments. A stock ranking based trading simulation is used for fitness evaluation and a probabilistic metric is employed to choose a solution from the Pareto Front. The experiments are performed on the constituent stocks of the S&P BSE FMCG Index. The results, evaluated using RMSE, MEA and cumulative returns, are very promising.

Keywords: Portfolio optimization
Dynamic multi-objective optimization · Evolutionary algorithms
Genetic programming

1 Introduction

One of the primary goals of asset management is to distribute wealth optimally amongst various financial assets. Given the large number of possible permutations of building a portfolio of stocks, constructing an optimal portfolio is a challenging task. Traditionally financial analysts aim at minimizing the risk and maximizing the returns of a portfolio, hence making portfolio optimization a classic multi-objective problem. Markowitz introduced his famous Modern Portfolio Theory (MPT) [1] which advocates the concept of finding an ideal trade-off between return and risk.

Given the multi-objective nature of the problem, a stochastic based method like Evolutionary Algorithm (EA) is often applied. EA falls under the umbrella of probabilistic search heuristics that are analogous to natural selection and genetics. In EA paradigms a population of individuals undergo evolution using operators like crossover and mutation. Usually EA models allow the fitter individuals to reproduce more than the ones who are weak. Eventually all individuals of the population tend to converge towards the fittest individuals. Selection techniques are employed to choose the individuals which are allowed to mate. Generally,

EAs presume that the environments are static in nature i.e. the conditions in which they are evolved are similar to the conditions in which they are used. However in most applications, the environment is dynamic. Hence, the fittest individuals in one environment do not perform well in other environments.

Financial markets tend to exhibit cyclic trends of bull and bear phases which last for varying time spans, making the market highly unpredictable. Due to this erratic nature of financial markets, when evolutionary algorithms are applied to solve the problem of portfolio optimization, the individuals which perform well in one environment do not continue doing so in other environments. In this work, a novel approach for solving the problem of portfolio optimization in changing environments is proposed.

The rest of this paper is organized as follows. Section 2 presents an overview of multi-objective optimization techniques employed in dynamic environments. Section 3 presents a detailed study of the portfolio optimization problem. Section 4 contains our proposed MemSPEAII algorithm. Section 5 presents the details of the experimental setup. The comparative study and discussion is done in Sect. 6. Lastly, Sect. 7 summarizes the paper.

2 Related Works

The Modern Portfolio Theory proposed by Markowitz [1], was a breakthrough in the field of asset optimization. However, the theory has been subject to criticism due to its irrational assumptions such as unlimited short sell. In an attempt to address such drawbacks, a number of changes to the traditional model were suggested. Many researchers proposed alternative risk measure like semi-covariance. Moreover, the theory is also restated as a single-objective problem where the risk and return optimization is represented as a performance measure like Sortino Ratio [2], Sharpe Ratio [3,4], Treynor ratio [5], Sterling ratio [5] or something similar. In [6], the problem was solved as a tri-objective optimization problem where the number of stocks in the portfolio was considered as a third objective along with return and risk. In [7,8], the transaction lots are minimized, hence minimizing the transaction cost. Along with research in modifying the objective functions, some work has also been done in adding additional constraints like cardinality constraints [9].

EAs are used as a classic solution to the problem of asset optimization [10–12]. An EA paradigm was introduced in [13] which approached the problem of portfolio optimization in a practical context by including pragmatic constraints in the model. In [14], the envelope-based MOEA, which integrated an active set algorithm with a MOEA, was suggested. In [15], several MOEAs, such as, Non-dominated Sorting Genetic Algorithm II (NSGA-II), Niche Pareto Genetic Algorithm 2 (NPGA-II), Pareto Envelope-based Selection Algorithm (PESA), and e-Multi-objective Evolutionary Algorithm (e-MOEA), Strength Pareto Evolutionary Algorithm 2 (SPEA-II), were applied to the portfolio optimization problem. Although abundant work has been done in the application of EA in finance, most of the work has been done with respect to evolution in static environments. The work done in [16] examines the efficacy of MOGP for dynamic

environments in portfolio optimisation. Moreover, it introduces a new performance and statistical measure called robustness.

In recent years, researchers have done extensive work on the problem of dynamic multi-objective optimization (MOO). The methods utilized for dynamic MOO range from simple alterations in the GP parameters [17] to maintaining diversity in the population. In [18] a part of the population is re-initialized whenever a change in the environment is detected to maintain diversity. The re-initialized random population of that generation is termed as “immigrants” and they are responsible to enhance and control the diversification of the population in order to avoid fast convergence. In [19] a hybrid paradigm formed by combining DNSGA-II-A and DNSGA-II-B is suggested for portfolio optimization. The worst individuals were replaced by random individuals and the mutation rate was increased to enhance diversity.

3 Portfolio Optimization

A portfolio is a collection of stocks, which provides diversification and therefore protects the investor against the price volatility of the underlying equities. In general, portfolio optimization is the process of selecting a group of optimal stocks for the portfolio, and dividing the investor’s wealth in those stocks in certain proportions. The portfolios are represented by a N-vector $w = (w_1, w_2, \dots, w_N)$, where w_N represents the compositional weightage of the N th stock in the portfolio. The weights w need to satisfy the following equation:

$$\sum_{i \in N} w_i = 1 \tag{1}$$

The return of the stocks are represented by the vector- r such that $r = (r_1, r_2, \dots, r_N)$ and the expected returns are denoted by $\mu = (\mu_1, \mu_2, \dots, \mu_N)$. The $N * N$ covariance matrix is computed as shown in Eq.2. The symbol σ_{ij} denotes the covariance between the asset i and j .

$$\Sigma = \begin{pmatrix} \sigma_{11} & \dots & \sigma_{1N} \\ \dots & \dots & \dots \\ \sigma_{N1} & \dots & \sigma_{NN} \end{pmatrix} \tag{2}$$

The goal of the multi-objective portfolio optimization problem is to find a trade-off between risk and return of portfolio p as depicted in Eqs.3 and 4.

$$\text{maximize } \mu_p = w^T \mu \tag{3}$$

$$\text{minimize } \sigma_p^2 = w^T \Sigma w \tag{4}$$

The solution to Eqs.3 and 4 forms a set of points that constitute the efficient frontier. Each point on the efficient frontier represents a portfolio that yields a maximum expected return for a certain amount of variance.

4 Proposed Method

The goal of EAs that are employed in portfolio optimization is to obtain a Pareto Front, which tends to approximate the Efficient Frontier accurately. Given the erratic nature of financial markets, the Pareto Front solutions developed for one environment do not remain optimal for a new environment. This paper proposes a solution to the problem of multi-objective optimization in dynamic environments by enhancing the classic SPEAII [20] with the use of memory. SPEAII has been embedded with the ability to store good solutions from all environments in an explicit memory. Essentially, the memory stores solutions that performed exceptionally well across all previous market conditions. The memory solutions are used to re-initialize a part of the population when an environment change occurs. With the help of memory, the algorithm explores better areas in the search space. In the absence of memory, the model starts by exploring random areas in the search space which deteriorates its searching capability. The following sections discuss the evolutionary framework of the model, the trading architecture and the proposed algorithm in detail.

4.1 Evolution Architecture

The model uses genetic programming which is a subset of EA wherein the individuals are represented in tree structure. The leaves of the tree are called terminal set which comprise of the input provided to the GP tree. The non-terminal nodes of the tree consist of the operators which process the input and generate an output. In this work, the terminal nodes are represented by the financial factors such as price, volume, moving average, etc. and the non-terminal nodes are represented by operators such as addition, subtraction, multiplication, etc. Each tree represents a formula formed by permutation of financial factors and operators and such a tree is called factor model. When each stock's financial factors are given as an input to the tree, it returns a processed value known as the factor model value for that stock. The model uses a MOGP approach in which the evolution of the GP Trees is governed by two conflicting objectives.

MOGP explores the search space for finding an optimal trade-off between multiple objectives. In a search space ω with n decision parameters, a solution is represented by the vector $x = [x_1, x_2, \dots, x_n]$. The values of this vector are changed to traverse ω in order to find optimal solutions. The multi-objective optimization problem is to find the value of vector x which are a solution to $F(x) = [f_1(x), f_2(x), \dots, f_k(x)]$ where k is the total number of objective functions.

In case of a maximization problem, a solution vector $q = [q_1, q_2, \dots, q_n]$ is said to dominate another solution vector $p = [p_1, p_2, \dots, p_n]$ if and only if Eqs. 5 and 6 hold true.

$$f_i(q) \geq f_i(p) \text{ for all } i \in [1, 2, \dots, k] \quad (5)$$

$$f_j(q) > f_j(p) \text{ for at least one } j \in [1, 2, \dots, k] \quad (6)$$

The solutions which are not dominated by any other feasible set of solutions that would improve some objective function without deteriorating another are termed as Pareto Optimal Solutions. The plot of Pareto Optimal Solutions in the value space of the objective functions is called as the Pareto Optimal front. The mathematical equation representing the Pareto Optimal Set and Pareto Optimal front is given by Eqs. 7 and 8.

$$P' = \{x : x \in \omega \mid \nexists q \in \omega : F(q) \geq F(x)\} \tag{7}$$

$$PF' = F(x) = \{f_1(x), f_2(x), \dots, f_n(x) \mid x \in P'\} \tag{8}$$

4.2 Trading Architecture

The outline of the trading architecture is given in Fig. 1. The system consists of a simulation of an investment strategy, as well as an embedded MOGP for making the trading decisions. In the trading simulation, stocks are traded based on the buy and sell decisions made by an individual (GP Tree). The accuracy of the decisions governs the fitness value of that individual. At the start of every month, each individual generates η factor model values corresponding to η stocks, where η is the total number of stocks in the stock universe. These η stocks are ranked according to their factor model values, where the stock with a lower factor model value is assigned a higher rank. The simulator buys the stocks in the top quartile and sells the ones which are not in the top quartile. Accordingly these buy and sell decision generate return R_m for that month. This process is repeated for each month, for a period of one year. The monthly return R_m is used to calculate the annual return and variance of the portfolio using Eqs. 9 and 10 respectively.

$$\text{Annual Return} = \prod_{m=1}^{12} (1 + R_m) \tag{9}$$

$$\text{Variance} = \sqrt{\frac{\sum_{m=1}^{12} (R_m - \frac{\sum_{m=1}^{12} R_m}{n})^2}{n}} * \sqrt{12} \tag{10}$$

Since the individual dictates the portfolio composition over the period of one year, it can be conclude that an individual is analogous to a portfolio. The annual return and variance of a portfolio are used as the fitness values of the individual.

4.3 Memory Enhanced SPEAII

The outline of MemSPEAII algorithm is given in Fig. 2. The following sections discuss the basic structure of the evolution, the paradigm for updating the memory and the steps taken by the model when a new environment occurs.

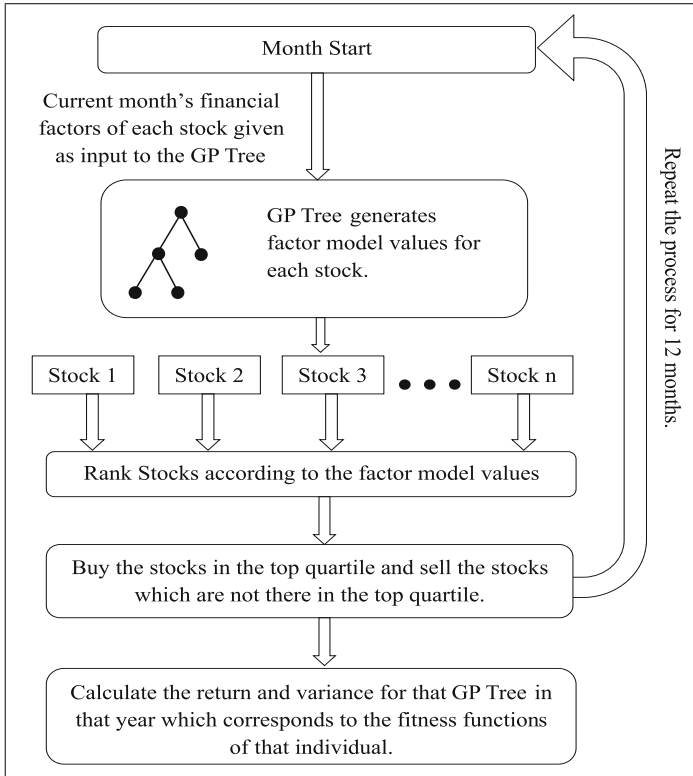


Fig. 1. Trading simulation

Evolution for an Environment. When the model is trained for its first environment, the algorithm begins by randomly initializing the main population. The main population undergoes mating using mutation and crossover operators to generate offspring. Later on, the individuals in the main population and the offspring are both evaluated. The trading simulation discussed in Sect. 4.2 is used to evaluate the fitness value of an individual i.e. risk and annual return. The non-dominated individuals are selected from the main population and offspring using SPEAII. The selected individuals are then sent to the next generation where they start off as the main population. The individuals in the population continue evolving for the specified number of generations to complete the evolution for that environment.

Updating the Memory. The memory starts empty and it is updated after every z generation, where z is the memory update frequency. To find new peaks and promote diversity, a new randomly initialized population is introduced which is known as the search population. The individuals in the main population as well as the search population are the probable candidates to be stored in the memory.

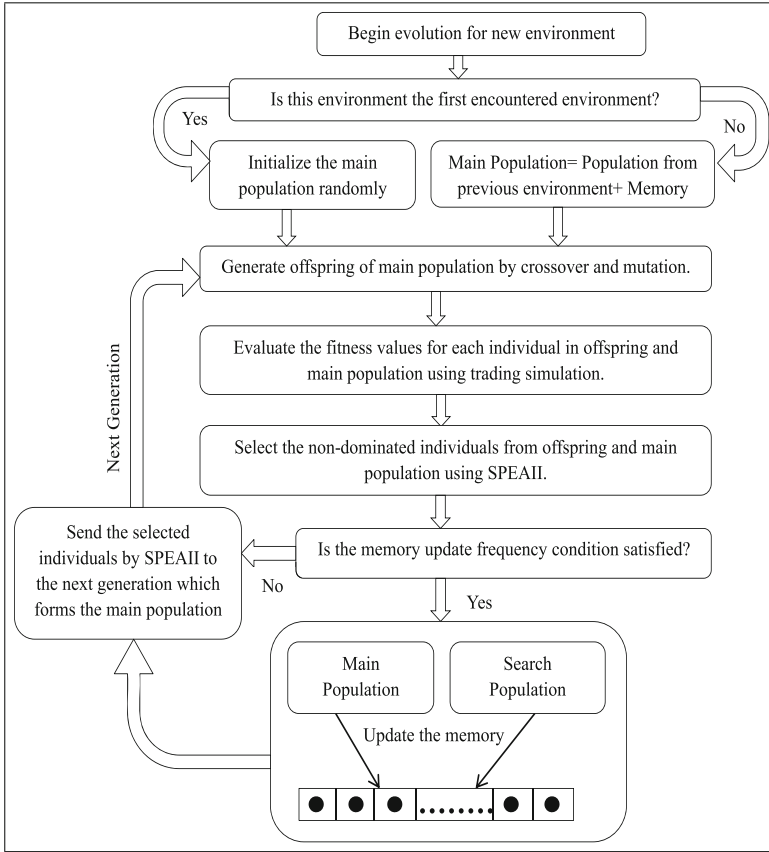


Fig. 2. MemSPEAII algorithm

For updating the memory, the non-dominated individuals are selected from the main and search population. The non-dominated solutions are directly stored in the memory until the memory gets completely filled. When the memory becomes full, the nearest non-dominated individual to each memory element is searched using Euclidean distance. The memory solution is replaced by the nearest non-dominated solution, only if the nearest non-dominated solution is better than the memory solution. Otherwise the memory remains unchanged. The metric used for determining the better individual is the probability of outperforming the equal weighted index discussed in Sect. 4.4. The individual with a higher probability of beating the equal weighted index is stored in the memory.

When the Environment Change Occurs. Whenever an environment change occurs, the algorithm starts a fresh evolution for the new environment. The individuals stored in the memory are combined with the main population at the beginning of the fresh evolution.

Algorithm 1. Pseudo-code for Updating the Memory

```

1: procedure UPDATEMEMORY
2:    $P \leftarrow$  Main Population
3:    $Q \leftarrow$  Search Population
4:    $M \leftarrow$  Memory Population
5:    $K \leftarrow$  Size of Memory
6:    $nonDominated \leftarrow FindNonDominated(P \cup Q)$ 
7:   if memory is empty then
8:      $M \leftarrow$  Fill with first K individuals of  $nonDominated$ 
9:   for each  $m$  in  $M$  do
10:     $C \leftarrow FindNearest(m, nonDominated)$ 
11:    if  $Probability(C) > Probability(m)$  then
12:       $ReplaceInMemory(m, C)$ 

```

4.4 Choosing from Evolved Pareto Front

Once the evolved Pareto Optimal Front is obtained, an individual from the Pareto front is to be chosen for testing. A widely used method for picking an optimal individual from a Pareto is the Knee Point Driven Evolutionary Algorithm [21], in which the middle individual of the Pareto Optimal Front is chosen. In this work a novel heuristic based parameter for choosing is introduced. Each individual is associated with a probability metric, which is defined as the number of months it outperforms the equal weighted index in terms of returns. The equal weighted index is formed by buying equal proportions of all the stocks in the stock universe. The probability metric is calculated as shown in Eq. 11. The individual with the highest probability is used for testing.

$$Score(j, individual) = \begin{cases} 1 & \text{if } individual \text{ outperforms index in month } j \\ 0 & \text{otherwise} \end{cases}$$

$$Probability(individual) = \frac{\sum_{month=1}^N Score(month, individual)}{N} \quad (11)$$

5 Experimental Setup

5.1 Data and Parameters

In this paper, the constituent stocks of the S&P BSE FMCG Index serves as the universe of stocks. The transaction period was considered from January 2007 to December 2016 with the training data spanning from January 2007 to December 2013. The model was tested from January 2014 to December 2016. In this work the environment was defined as one year and the beginning of a new year corresponded to the change in environment [16].

Genetic programming trees consisting of terminal and non-terminal nodes are constrained to the maximum depth of 17 to avoid the bloating problem [22]. The operators which form the non-terminal nodes are addition, subtraction, division, multiplication, power 2 and power 3. The financial factors which form the terminal nodes of the GP Trees include 19 features consisting technical, fundamental and macro-economic factors. Some of such financial factors include Price, Price to Earnings ratio, Inventory Turnover Ratio, 15 Day Moving Average, etc. [16] The data collected is monthly and it is normalized such that it does not exhibit forward biasing [16]. The mutation probability and the crossover probability were set to 0.1 and 0.8 respectively. The method used for random tree generation was Ramped half and half [22]. The main population size was set to 500 and the number of generations chosen for each environment was 50. The size of the search population was set equal to the size of the main population, while the size of the explicit memory was set to 50. The memory update frequency was set to 10 generations.

5.2 Walk Forward Testing

The model used walk forward testing and was initially trained from the year 2007 till the year 2013. The population evolved for the environment of 2013 was used for testing in the unseen environment of 2014. After completing the testing of 2014, the training window is shifted forward by one year. Hence the model trains in 2014 and is tested in the subsequent environment of 2015. Likewise, walk forward testing is continued for the entire testing period.

5.3 Comparative Study

The probability based MemSPEAII is compared with four other models namely knee point based MemSPEAII, NSGAII [23], random ranking and Buy and Hold index. In knee Point based MemSPEAII, the central individual on the Pareto Front is used for testing. The knee point based model is used to compare probability as a metric for choosing a solution from Pareto. NSGAII is a widely used MOEA algorithm which was used to gauge the performance of MemSPEAII. Random ranking was used as a control experiment in which stocks were ranked randomly. The Buy and Hold Index formed by buying one stock of every company, is a widely used benchmark in asset management.

6 Results and Discussions

Although the model's explicit objective is maximizing the returns and minimizing the risk, the inherent goal of the model is to rank the stocks accurately. Thus, two commonly used performance evaluation metrics are employed to evaluate the rank prediction capability of the model. The metrics used are Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) and the results are shown in Table 1. The results depict that the MemSPEAII model outperforms other

Table 1. Model comparison based on accuracy of rank prediction

Model	RMSE	MAE
Probability based MemSPEAII	3.56	2.60
Knee point based MemSPEAII	3.77	2.94
NSGAII	4.27	3.31
Random ranking	4.57	3.70

models and establishes the effectiveness of memory enhancement in evolution, and probability metric in selection.

In the models not enhanced with memory, the optimization algorithm has to extensively search large portions of the search space and hence evolution to the optimal solutions becomes cumbersome. However in MemSPEAII, the memory guides the search process by providing nearly optimal solutions at the start of a new evolution. Hence the population guided by memory converges faster as shown in Fig. 3. In the same number of generations, the memory enhanced version formed a better Pareto Optimal Front than the algorithm with no memory.

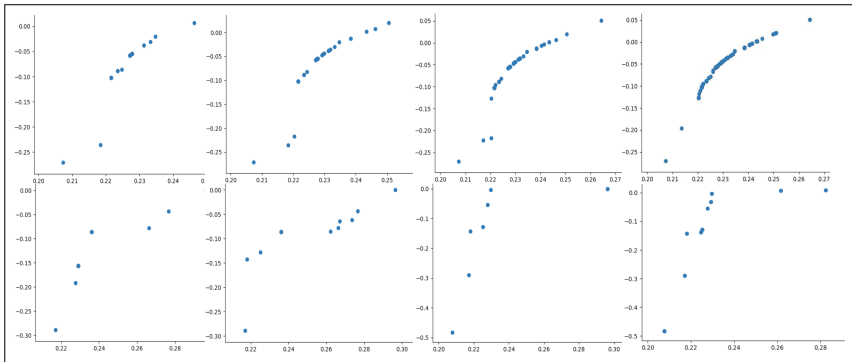


Fig. 3. Pareto evolution comparison of memory enhanced (top) with traditional EA (bottom)

The walk forward testing results obtained for the years 2014, 2015 and 2016 have been shown in the Fig. 4. It can be clearly inferred that when MemSPEAII used probability for testing, it produced exceptional returns of 405.92% over the period of 3 years, which outperformed all other models. Over the same years, knee point based MemSPEAII, NSGAII and Buy and Hold yielded cumulative returns of 335%, 144% and 115% respectively. The control experiment using random ranking gave 85% returns over the same period. The year 2016 was a challenging year for all models because only a few stocks in the stock universe exhibited an uptrend. The models would have had to rank the stocks with very high precision

to avoid generating losses. NSGAII gave -6% return in the year 2016. However, both the memory based models i.e. probability based MemSPEAII and knee point based MemSPEAII picked stocks diligently hence giving returns of 47.5% and 84% respectively in 2016. The stock ranking in these models was accurate by virtue of the enhancement provided by the explicit memory during evolution.

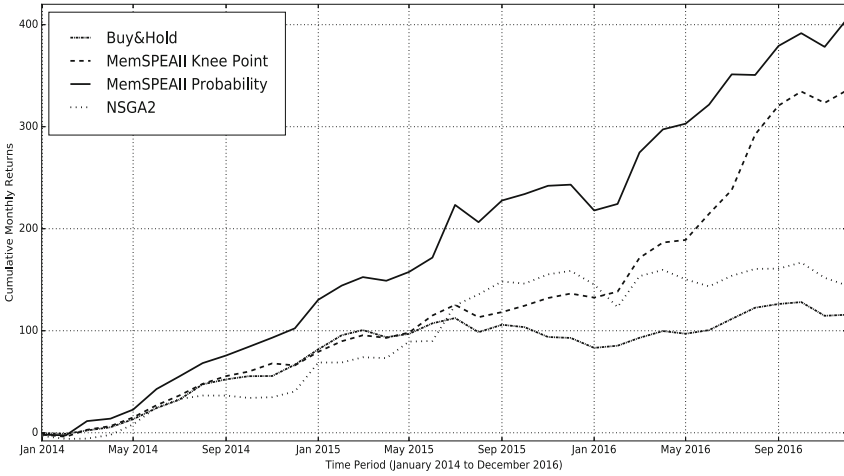


Fig. 4. Cumulative returns from January 2014 to December 2016

The probability based approach used for choosing a solution from Pareto Optimal Front showed exceptional results. Table 2 shows the cumulative returns of the individual with the highest probability and the individual with lowest probability of beating the index. As observed in the Table 2, the individual with highest probability of beating the index convincingly outperformed the individual with the lowest probability.

Table 2. Performance analysis of probability based MemSPEAII

Selection criteria	2014	2014–2015	2014–2016	Probability
Highest probability	102.02%	241.91%	405.92%	0.84
Lowest probability	32.14%	49.91%	162.34%	0.57

7 Conclusion

In this paper, a novel memory-based algorithm is proposed and is implemented as a solver for portfolio optimization problem. The proposed algorithm MemSPEAII stores a number of non-dominated solutions in an explicit memory and

whenever an environment change occurs, a part of the population is initialized with the memory solutions. The trading simulation was performed on the constituent stocks of S&P BSE FMCG Index. To choose an optimal solution from the Pareto, the concept of probability was employed. The returns generated by probability based MemSPEAII was compared with knee point based MemSPEAII, NSGAII and Buy&Hold index. The results show that the probability based MemSPEAII convincingly outperformed other models by yielding a cumulative return of 405.92% over a period of 3 years.

However, further work is required to validate the model on a different stock universe and a longer time period. Also, the evolutionary algorithms' performance in dynamic portfolio optimization must be compared with other mathematical approaches like quadratic programming.

Acknowledgement. The authors would like to thank Prof. (Mrs.) Lakshmi Kurup, D. J. Sanghvi College of Engineering, for her assistance on the machine learning front and Dr. Ram Mangrulkar, D. J. Sanghvi College of Engineering, for his review and comments that greatly improved the manuscript.

References

1. Markowitz, H.: Portfolio selection. *J. Financ.* **7**(1), 77–91 (1952)
2. Sortino, F.A., Price, L.N.: Performance measurement in a downside risk framework. *J. Invest.* **3**(3), 59–64 (1994)
3. Sharpe, W.F.: The sharpe ratio. *J. Portfolio Mgmt.* **21**(1), 49–58 (1994)
4. Sharpe, W.F.: Capital asset prices: a theory of market equilibrium under conditions of risk. *J. Financ.* **19**(3), 425–442 (1964)
5. Bacon, C.R.: *Practical Portfolio Performance Measurement and Attribution*, vol. 568. Wiley, Hoboken (2011)
6. Anagnostopoulos, K.P., Mamanis, G.: A portfolio optimization model with three objectives and discrete variables. *Comput. Oper. Res.* **37**(7), 1285–1297 (2010)
7. Mansini, R., Speranza, M.G.: Heuristic algorithms for the portfolio selection problem with minimum transaction lots. *Eur. J. Oper. Res.* **114**(2), 219–233 (1999)
8. Lin, C.C., Liu, Y.T.: Genetic algorithms for portfolio selection problems with minimum transaction lots. *Eur. J. Oper. Res.* **185**(1), 393–404 (2008)
9. Chang, T.J., Meade, N., Beasley, J.E., Sharaiha, Y.M.: Heuristics for cardinality constrained portfolio optimisation. *Comput. Oper. Res.* **27**(13), 1271–1302 (2000)
10. Tapia, M.G.C., Coello, C.A.C.: Applications of multi-objective evolutionary algorithms in economics and finance: a survey. In: *IEEE Congress on Evolutionary Computation, CEC 2007*, pp. 532–539. IEEE (2007)
11. Coello, C.A.C.: Evolutionary multi-objective optimization and its use in finance. In: *Handbook of Research on Nature Inspired Computing for Economy and Management*. Idea Group Publishing, Hershey (2006)
12. Ponsich, A., Jaimes, A.L., Coello, C.A.C.: A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. *IEEE Trans. Evol. Comput.* **17**(3), 321–344 (2013)
13. Chiam, S., Tan, K., Al Mamum, A.: Evolutionary multi-objective portfolio optimization in practical context. *Int. J. Autom. Comput.* **5**(1), 67–80 (2008)

14. Branke, J., Scheckenbach, B., Stein, M., Deb, K., Schmeck, H.: Portfolio optimization with an envelope-based multi-objective evolutionary algorithm. *Eur. J. Oper. Res.* **199**(3), 684–693 (2009)
15. Anagnostopoulos, K.P., Mamanis, G.: The mean-variance cardinality constrained portfolio optimization problem: an experimental evaluation of five multiobjective evolutionary algorithms. *Expert Syst. Appl.* **38**(11), 14208–14217 (2011)
16. Hassan, G., Clack, C.D.: Robustness of multiple objective GP stock-picking in unstable financial markets: real-world applications track. In: *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, pp. 1513–1520. ACM (2009)
17. Cobb, H.G.: An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments. Technical report, Naval Research Lab Washington D.C. (1990)
18. Yang, S., Tinós, R.: A hybrid immigrants scheme for genetic algorithms in dynamic environments. *Int. J. Autom. Comput.* **4**(3), 243–254 (2007)
19. Filipiak, P., Lipinski, P.: Dynamic portfolio optimization in ultra-high frequency environment. In: Squillero, G., Sim, K. (eds.) *EvoApplications 2017*. LNCS, vol. 10199, pp. 34–50. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-55849-3_3
20. Laumanns, M.: SPEA2: Improving the strength Pareto evolutionary algorithm. Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK) (2001)
21. Zhang, X., Tian, Y., Jin, Y.: A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **19**(6), 761–776 (2015)
22. Koza, J.: *Genetic Programming on the Programming of Computers by Means of Natural Selection*. The MIT Press, Cambridge (1992)
23. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)

Author Index

- Abhuri, Harika 231
Agrawal, Ruchit 287
Ajith Kumar, I. 371
Ali, Lasker Ershad 168
Anand, Tanushree 335
Annapurna, H. 133
Avinash, J. 371
- Banerjee, Swati 212
Bhardwaj, Basant 81
Bhattacharya (Halder), Sharmistha 412
Boya, Sravani 308
- Carneiro, Brian 90
Chakraborti, Sutanu 250, 274
Chakraborty, Konika 1
Chakraverty, Shampa 402
Chennuru, Venkata Krishnaveni 43
Cho, Sung-Bae 144
Chug, Stuti 64
Cobos, Carlos Alberto 198
Corrales, Juan Carlos 198
- Das, Bappaditya 391
Das, Dipankar 391
Debnath, Kalyani 412
Debnath, Soumyadeep 391
Desarkar, Maunendra Sankar 359
Dey, Somnath 111, 156
Drias, Habiba 381
Dwivedi, Rudresh 111
- Gangashetty, Suryakanth V. 231
Ghosh, Ashish 1, 144
Ghosh, Kuntal 212
Ghosh, Shinjini 73
Ghosh, Susmita 144, 297
Ghoshal, Bibhas 335
Gupta, Shubhrata 100
Guru, D. S. 133
Gurugubelli, Krishna 189
- Hazarika, Shyamanta M. 81
- Jadon, Mukesh 64
Jebakumari Beulah Vasanthi, J. 178
Jothi, R. 35
- Kant, Vibhor 64
Kathirvalavakumar, T. 178
Kavuluru, Ramakanth 22
Khapra, Mitesh 250
Kukreti, Yash 402
Kumar, K. R. Prasanna 221
Kumar, M. Anand 320
Kumar, Nagendra 308
Kumar, Ravi Kant 11
Kumar, S. Sachin 320
- Labhishetty, Sahiti 274
Law, Anwasha 1
- Ma, Jinwen 168
Madisetty, Sreekanth 359
Mandal, Anupam 221
Manjunatha, K. S. 133
Maysuradze, Archil 123
Mesteskiy, Leonid 54
Mishra, Shrija 11
Misra, Dipti 287
Mitra, Pabitra 221
Moulai, Hadjer 381
- Nagipogu, Rajivteja 274
Newaz, Sarfaraz 168
- Oswald, C. 371
- Pal, Utpal 412
Patel, Jay 261
Peixoto, Rui 90
Petrovskiy, Mikhail 240
Portela, Filipe 90
Pospelova, Irina 240
Prasada, Geeta Ramani Bala 11

- Prasath, Rajendra 231
Pulugandla, Bhargav 189
Pushp, Sumant 81
- Rai, Sunny 402
Ranabothu, Neha 308
Rehkaab, Nourelhouda 381
Ripon, Kazi Shah Nawaz 168
Roy, Rahul 144, 297
- S Chandran, Keerthi 212
Saha, Baishali 335
Santos, Manuel Filipe 90
Sanyal, Goutam 11
Shah, Priyank 424
Shah, Sanket 424
Sharma, Anurag 335
Sharma, Ram Prakash 156
Shekar, B. H. 54, 123
Shekhar, Mihir 287
Shrivastava, Manish 189, 231
Shrivastava, Shashank 250
Siddiqi, Ayesha 274
Sierra Martínez, Luz Marina 198
- Singh, Manish 308
Singh, Vikram 261
Sivaselvan, B. 371
Soman, K. P. 320
Sreenivasulu, Madichetty 348
Sridevi, M. 348
Subba, Prannoy 297
Suryamukhi, K. 308
Suvorov, Mikhail 123
- Tayal, Devendra K. 402
Thakur, Kavita 100
Thakur, Vikrant Singh 100
Timmappareddy, Sobha Rani 43
Tran, Tung 22
Tsarev, Dmitry 240
- Vegesna, Vishnu Vidyadhara Raju 189
Vuppala, Anil Kumar 189
Vydana, Hari Krishna 189
- Yadandla, Anusha 308