Ying Tan
Yuhui Shi
Milan Tuba (Eds.)

Communications in Computer and Information Science 1234

Data Mining and Big Data

5th International Conference, DMBD 2020
Belgrade, Serbia, July 14–20, 2020
Proceedings

Springer

Communications in Computer and Information Science 1234

More information about this series at http://www.springer.com/series/7899

Ying Tan · Yuhui Shi · Milan Tuba (Eds.)

Data Mining and Big Data

5th International Conference, DMBD 2020 Belgrade, Serbia, July 14–20, 2020 Proceedings

Springer

Editors
Ying Tan
Peking University
Beijing, China

Milan Tuba
Singidunum University
Belgrade, Serbia

Yuhui Shi
Southern University of Science
and Technology
Shenzhen, China

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-981-15-7204-3 ISBN 978-981-15-7205-0 (eBook)
https://doi.org/10.1007/978-981-15-7205-0

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, expressed 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.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721,
Singapore

Preface

This volume (CCIS vol. 1234) constitutes the proceedings of the 5th International Conference on Data Mining and Big Data (DMBD 2020) which was held virtually online during July 14–20, 2020.

The DMBD conference serves as an international forum for researchers and practitioners to exchange latest advantages in theories, technologies, and applications of data mining and big data. The theme of DMBD 2020 was "Serving Life with Data Science." DMBD 2020 was the fifth in the conference series, following successful events in Bali, Indonesia (DMBD 2016), Fukuoka, Japan (DMBD 2017), Shanghai, China (DMBD 2018), and Chiang Mai, Thailand (DMBD 2019).

Data mining refers to the activity of going through big data sets to look for relevant or pertinent information. This type of activity is a good example of the axiom "looking for a needle in a haystack." The idea is that businesses collect massive sets of data that may be homogeneous or automatically collected. Decision makers need access to smaller, more specific pieces of data from these large sets. They use data mining to uncover the pieces of information that will inform leadership and help chart the course for a business. Big data contains a huge amount of data and information and is worth researching in depth. Big data, also known as massive data or mass data, refers to the amount of data involved that are too great to be interpreted by a human. Currently, the suitable technologies include data mining, crowdsourcing, data fusion and integration, machine learning, natural language processing, simulation, time series analysis, and visualization. It is important to find new methods to enhance the effectiveness of big data. With the advent of big data analysis and intelligent computing techniques, we are facing new challenges to make the information transparent and understandable efficiently. DMBD 2020 provided an excellent opportunity as an academic forum for researchers and practitioners to present and discuss the latest scientific results, and innovative ideas and advantages in theories, technologies, and applications in data mining and big data. The technical program covered many aspects of data mining and big data as well as intelligent computing methods applied to all fields of computer science, machine learning, data mining and knowledge discovery, data science, etc.

DMBD 2020 was originally planned to be held at Singidunum University, Serbia, but after carefully evaluating most announcements and guidance regarding COVID-19, as well as restrictions on overseas travel released by relevant national departments, the DMBD 2020 Organizing Committee made the decision to host DMBD 2020 as a virtual conference, keeping the scheduled dates of July 14–19, 2020. The DMBD 2020 technical team provided the ability for the authors of accepted papers to present their work through an interactive online platform or video replay. The presentations by accepted authors will be made available to all registered attendees online.

DMBD 2020 received 39 submissions and invited manuscripts from about 91 authors in 11 countries and regions (Brunei, China, Colombia, Cuba, India, Japan, Malaysia, Russia, South Korea, Thailand, and Venezuela). Each submission was reviewed by at least 2 reviewers, and on average 3.1 reviewers. Based on rigorous reviews by the Program Committee members and reviewers, 10 high-quality papers were selected for publication in this proceedings volume with an acceptance rate of 25.64%. The contents of these papers cover major topics of data mining and big data.

On behalf of the Organizing Committee of DMBD 2020, we would like to express our sincere thanks to the International Association of Swarm and Evolutionary Intelligence (IASEI)(iasei.org) for its sponsorship, to Peking University, Southern University of Science and Technology, and Singidunum University for their co-sponsorship, and to Computational Intelligence Laboratory of Peking University and IEEE Beijing Chapter for its technical cosponsorship, as well as to our supporters of International Neural Network Society, World Federation on Soft Computing, Beijing Xinghui Hi-Tech Co., and Springer Nature.

We would also like to thank the members of the Advisory Committee for their guidance, the members of the International Program Committee and additional reviewers for reviewing the papers, and the members of the Publications Committee for checking the accepted papers in a short period of time. We are particularly grateful to the proceedings publisher Springer for publishing the proceedings in the prestigious series of *Communications in Computer and Information Science*. Moreover, we wish to express our heartfelt appreciation to the plenary speakers, session chairs, and student helpers. In addition, there are still many more colleagues, associates, friends, and supporters who helped us in immeasurable ways; we express our sincere gratitude to them all. Last but not the least, we would like to thank all the speakers, authors, and participants for their great contributions that made DMBD 2020 successful and all the hard work worthwhile.

June 2020

Ying Tan
Yuhui Shi
Milan Tuba

Organization

General Co-chairs

Ying Tan	Peking University, China
Milan Tuba	Singidunum University, Serbia

Program Committee Chair

Yuhui Shi	Southern University of Science and Technology, China

Advisory Committee Chairs

Milovan Stanisic	Singidunum University, Serbia
Russell C. Eberhart	IUPUI, USA
Gary G. Yen	Oklahoma State University, USA

Technical Committee Co-chairs

Haibo He	University of Rhode Island, USA
Kay Chen Tan	City University of Hong Kong, China
Nikola Kasabov	Aukland University of Technology, New Zealand
Ponnuthurai Nagaratnam Suganthan	Nanyang Technological University, Singapore
Xiaodong Li	RMIT University, Australia
Hideyuki Takagi	Kyushu University, Japan
M. Middendorf	University of Leipzig, Germany
Mengjie Zhang	Victoria University of Wellington, New Zealand

Plenary Session Co-chairs

Andreas Engelbrecht	University of Pretoria, South Africa
Chaoming Luo	University of Mississippi, USA

Invited Session Co-chairs

Andres Iglesias	University of Cantabria, Spain
Haibin Duan	Beihang University, China
Junfeng Chen	Hohai University, China

Special Sessions Chairs

Ben Niu	Shenzhen University, China
Yan Pei	University of Aizu, Japan
Qirong Tang	Tongji University, China

Tutorial Co-chairs

Junqi Zhang	Tongji University, China
Shi Cheng	Shanxi Normal University, China
Yinan Guo	China University of Mining and Technology, China

Publications Co-chairs

Swagatam Das	Indian Statistical Institute, India
Radu-Emil Precup	Politehnica University of Timisoara, Romania

Publicity Co-chairs

Yew-Soon Ong	Nanyang Technological University, Singapore
Carlos Coello	CINVESTAV-IPN, Mexico
Yaochu Jin	University of Surrey, UK
Rossi Kamal	GERIOT, Bangladesh
Dongbin Zhao	Institute of Automation, Chinese Academy of Sciences, China

Finance and Registration Chairs

Andreas Janecek	University of Vienna, Austria
Suicheng Gu	Google Corporation, USA

Local Arrangement Chairs

Mladen Veinovic	Singidunum University, Serbia
Nebojsa Bacanin	Singidunum University, Serbia
Eva Tuba	Singidunum University, Serbia

Conference Secretariat

Maiyue Chen	Peking University, China

Program Committee

Miltos Alamaniotis	University of Texas at San Antonio, USA
Nebojsa Bacanin	Singidunum University, Serbia
Carmelo J. A. Bastos Filho	University of Pernambuco, Brazil

Tossapon Boongoen	Mae Fah Luang University, Thailand
David Camacho	Universidad Politcnica de Madrid, Spain
Abdelghani Chahmi	Universite des Sciences et Technologie d’Oran, Algeria
Vinod Chandra S. S.	University of Kerala, India
Hui Cheng	Liverpool John Moores University, UK
Jose Alfredo Ferreira Costa	UFRN, Brazil
Bei Dong	Shaanxi Nomal University, China
Qinqin Fan	Shanghai Maritime University, China
A. H. Gandomi Stevens	Stevens Institute of Technology, USA
Liang Gao	Huazhong University of Science and Technology, China
Shangce Gao	University of Toyama, Japan
Teresa Guarda	Universidad Estatal da Peninsula de Santa Elena (UPSE), Ecuador
Weian Guo	Tongji University, China
Weiwei Hu	Peking University, China
Dariusz Jankowski	Wrocław University of Science and Technology, Poland
Qiaoyong Jiang	Xi’an University of Technology, China
Mingyan Jiang	Shandong University, China
Chen Junfeng	Hohai University, China
Kalinka Kaloyanova	University of Sofia, Bulgaria
Vivek Kumar	Universit degli Studi di Cagliari, Italy
Bin Li	University of Science and Technology of China, China
Qunfeng Liu	Dongguan University of Technology, China
Wenjian Luo	Harbin Institute of Technology, China
Wojciech Macyna	Wrocław University of Science and Technology, Poland
Katherine Malan	University of South Africa, South Africa
Yi Mei	Victoria University of Wellington, New Zealand
Efrn Mezura-Montes	University of Veracruz, Mexico
Sheak Rashed Haider Noori	Daffodil International University, Bangladesh
Endre Pap	University Singidunum, Serbia
Mario Pavone	University of Catania, Italy
Yan Pei	University of Aizu, Japan
Somnuk Phon-Amnuaisuk	Universiti Teknologi Brunei, Brunei
Pramod Kumar Singh	ABV-IIITM Gwalior, India
Joao Soares	GECAD, Portugal
Hung-Min Sun	National Tsing Hua University, Taiwan
Yifei Sun	Shaanxi Normal University, China
Ying Tan	Peking University, China
Paulo Trigo	ISEL, Portugal
Eva Tuba	University of Belgrade, Serbia
Milan Tuba	Singidunum University, Serbia
Agnieszka Turek	Wrocław University of Science and Technology, Poland

Mladen Veinovi	Singidunum University, Serbia
Gai-Ge Wang	China Ocean University, China
Guoyin Wang	Chongqing University of Posts and Telecommunications, China
Yan Wang	The Ohio State University, USA
Ka-Chun Wong	City University of Hong Kong, China
Yingjie Yang	De Montfort University, UK
Peng-Yeng Yin	National Chi Nan University, Taiwan
Junqi Zhang	Tongji University, China
Jie Zhang	Newcastle University, UK
Xinchao Zhao	Beijing University of Posts and Telecommunications, China
Yujun Zheng	Zhejiang University of Technology, China
Dejan Zivkovic	Singidunum University, Serbia
Miodrag Zivkovic	Singidunum University, Serbia

Contents

Adaptive and Dynamic Knowledge Transfer in Multi-task Learning with Attention Networks

Tao Ma and Ying Tan(✉)

Key Laboratory of Machine Perception (MOE), Department of Machine Intelligence,
School of Electronics Engineering and Computer Science, Peking University,
Beijing 100871, China
{pku_mark,ytan}@pku.edu.cn

Abstract. Multi-task learning has shown promising results in many applications of machine learning: given several related tasks, it aims to generalize better on the original tasks, by leveraging the knowledge among tasks. The knowledge transfer mainly depends on task relationships. Most of existing multi-task learning methods guide learning processes based on predefined task relationships. However, the associated relationships have not been fully exploited in these methods. Replacing predefined task relationships with the adaptively learned ones may lead to superior performance as it can avoid the misguiding of improper predefinition. Therefore, in this paper, we propose Task Relation Attention Networks to adaptively model the task relationships and dynamically control the positive and negative knowledge transfer for different samples in multi-task learning. To evaluate the effectiveness of the proposed method, experiments on various datasets are conducted. The experimental results demonstrate that the proposed method outperforms both classical and state-of-the-art multi-task learning baselines.

Keywords: Representation learning · Multi-task learning · Knowledge transfer · Attention networks

1 Introduction

Multi-task learning (MTL) aims to generalize better on the original tasks, by leveraging the shared knowledge among tasks [15]. In MTL, positive knowledge transfer leads to improved performance, because: 1) more related information is incorporated into the target tasks, which benefits the feature learning process to obtain better feature representations; 2) the incorporated information of positively related tasks acts as regularizer to avoid the risks of over-fitting. The knowledge transformer mainly depends on the task relationships. Therefore, how to appropriately model task relationships and how to control the knowledge transfer among tasks are crucial in MTL.

Y. Tan et al. (Eds.): DMBD 2020, CCIS 1234, pp. 1–13, 2020.
https://doi.org/10.1007/978-981-15-7205-0_1

With the recent advances in deep learning, MTL with deep neural networks has been used widely and successfully across many applications of machine learning, from natural language processing [7] to computer vision [11]. In most of these existing methods, the task relationships guiding the learning process are generally predefined, and the knowledge transfer among tasks relies on the sharing of hidden layers. Despite they have achieved promising results, but there are still several challenges for further improving the performance of MTL methods.

The first challenge is adaptively learning the task relationships instead of relying on predefined relationships. For some multi-task learning problems with complex task associations, the pre-definition based on limited human knowledge requires costly human efforts. Besides, if there are not adequate efforts for sophisticated pre-definitions, there is likely to be negative knowledge transfer because of the misguiding of improperly predefined task relationships [8,16]. Since an improper pre-definition of task relationships may result in negative knowledge transfer, it is essential for MTL methods to adaptively and appropriately model the task relationships with learning-based modules.

The second challenge is controlling the knowledge transfer with the dynamically learned task relationships instead of the fixed ones. The relationships among tasks are not constantly fixed, but vary slightly from different samples. However, in most of the methods relying on pre-definition, the task relationships are fixed [9], even in some MTL methods equipped with adaptively learning modules [14]. A concrete example is, in the works of [21], the task relationships are determined by the inputs X and outputs Y of training data. In the testing phase, the model directly performs predictions on the learned relationships and the inputs X_{test}, i.e., the relationships are fixed for the testing data. However, the task relationships in different samples may not be necessarily consistent. Therefore, dynamically modeling these relationships based on different inputs may lead to superior performance.

To address these challenges, in this paper, we propose Task Relation Attention Networks (TRAN) to adaptively capture the task relationships and dynamically control the knowledge transfer in MTL. Specifically, TRAN is an attention-based model to adaptively capture the task relationships via task correlation matrix according to their inputs and specify the shared feature representations for different tasks. Since the task relationships are adaptively learned by TRAN during the learning process, it is replacing the predefined relationships. And TRAN relies on the inputs, therefore, it can dynamically model the correlations for different samples.

To evaluate the effectiveness of the proposed method, the experiments on various datasets are conducted. The experimental results demonstrate the proposed method can outperform both classical and state-of-the-art MTL baselines. The contributions of this paper can be summarized as follows:

- This paper proposes Task Relation Attention Networks (TRAN) to adaptively learn the task relationships to replace the predefined ones.
- The proposed method can dynamically control the knowledge transfer in multi-task learning based on the adaptively learned task relationships.

- This paper provides an explicable learning-based framework for multi-task learning to learn the shared feature representations for different tasks.

2 Related Works

2.1 Multi-task Learning

Multi-task Learning (MTL) provides an efficient framework for leveraging task relationships to achieve knowledge transfer and leads to improved performance.

The typical MTL architectures were sharing the bottom layers for all tasks and split top layers for different tasks, proposed by [3]. Afterwards, there have been some attempts to design partly shared architectures between tasks, instead of only sharing the bottom hidden layers. For leveraging the task relationships in MTL, there are some recent examples. The cross-stitch networks [14] learned an optimal combination of task-specific representations for each task using linear units. The tensor factorization model [20] generated the hidden layer parameters for each task. The multi-gate mixture-of-experts model [13] using gating mechanism to capture the task differences and implicitly model the task relationships. In the works of [21], they applied attention networks to statically capture the task relationships.

Compared to the typical MTL methods, these works performed better feature learning for shared and task-specific representations and achieved better performance. However, there are still some limitations: the method of [14] can hardly expand to a large number of tasks; the method of [20] relies on a specific application scenario; the method of [13] applies linear gates to implicitly model the task relationships and performs poor efficiency as the number of experts increases; the method of [21] captures the task relationships statically.

2.2 Attention-Based Neural Networks

The key idea of attention mechanism is mainly based on the human visual attention, which has been successfully applied in various applications, such as natural machine translation [12] and text summarization [1].

Graph attention networks [18] was proposed for feature learning of graph-structured data, based on the self-attention mechanism. [10] applied the self-attention networks for time series warping. [17] performed self-attention on sentence encoding models, called Transformer, dispensing with recurrence and convolutions entirely [17]. Based on the Transformer, a language representation model called BERT was proposed [5]. In this paper, we attempt to adaptively model the task relationships in MTL with self-attention networks. Based on the learned relationships, the knowledge transfer among tasks can be dynamically controlled and each task can obtain a better shared feature representation, leading the MTL method to better performance.

3 Method

3.1 Problem Statement for Multi-task Learning

Given a single task, which can be regression or classification, the formal definition is as follows:

$$\hat{\boldsymbol{Y}} = \mathrm{M}(\boldsymbol{X}) \approx \mathrm{E}\left(\boldsymbol{Y}|\boldsymbol{X}\right), \tag{1}$$

where $\boldsymbol{X}$ represents the inputs, $\boldsymbol{Y}$ represents the ground truth values, $\hat{\boldsymbol{Y}}$ is the predicted values, $\mathrm{E}(\cdot)$ is the mathematical expectation and $\mathrm{M}(\cdot)$ is the model.

In MTL, assuming that there are K tasks, the problem can be described as follows:

$$\begin{aligned}\left(\hat{\boldsymbol{Y}}_1, \hat{\boldsymbol{Y}}_2, ..., \hat{\boldsymbol{Y}}_K\right) &= \mathrm{MTL}\left(\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_K\right)\\ &\approx \mathrm{E}\left(\boldsymbol{Y}_1, \boldsymbol{Y}_2, ..., \boldsymbol{Y}_K|\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_K\right),\end{aligned} \tag{2}$$

where $\mathrm{MTL}(\cdot)$ is the multi-task learning method, $\boldsymbol{X}_i, \boldsymbol{Y}_i, \hat{\boldsymbol{Y}}_i$ are respectively the inputs, labels and predictions of each task. In some real-world scenarios, the inputs of different tasks can be the same, i.e., $\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_K = \boldsymbol{X}_s$.

3.2 Task Relation Attention Networks

We apply attention networks to model the task relationships to help shared feature learning, called Task Relation Attention Networks (TRAN). Given K tasks, the inputs are a set of task features, $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_K\}, \boldsymbol{X}_i \in \mathrm{R}^n$, n is the dimensionality of features.

Given task i and j, there is a shared attention network a_t measuring the attention correlations $\boldsymbol{e}_{ij}$ between two tasks, processed as follows:

$$\boldsymbol{e}_{ij} = a_t(\boldsymbol{W}_i\boldsymbol{X}_i||\boldsymbol{W}_j\boldsymbol{X}_j),\ i, j = 1, 2, ..., K, \tag{3}$$

where $\boldsymbol{W}_i \in \mathrm{R}^{n\times d}$ and $\boldsymbol{W}_j \in \mathrm{R}^{n\times d}$ represent the encoding networks, modeling the original inputs into high-level latent representations with the dimensionality of d, and $||$ is the concatenation operation.

The attention weights for task i are normalized by softmax function to obtain the associated relationships between other tasks and task i, $(\boldsymbol{\alpha}_{i1}, \boldsymbol{\alpha}_{i2}, ..., \boldsymbol{\alpha}_{iK})$, processed as:

$$\boldsymbol{\alpha}_{ij} = \mathrm{softmax}_i(\boldsymbol{e}_{ij}) = \frac{\exp(\boldsymbol{e}_{ij})}{\sum_{k=1}^{K}\exp(\boldsymbol{e}_{ik})}, j = 1, 2, ..., K. \tag{4}$$

The attention networks a_t are implemented with fully-connected neural networks with the activation function of LeakyReLU. And the learning process of attention networks can be described as:

$$\boldsymbol{\alpha}_{ij} = \frac{\exp\left[a_t(\boldsymbol{W}_i\boldsymbol{X}_i||\boldsymbol{W}_j\boldsymbol{X}_j)\right]}{\sum_{k=1}^{K}\exp\left[a_t(\boldsymbol{W}_i\boldsymbol{X}_i||\boldsymbol{W}_k\boldsymbol{X}_k)\right]}. \tag{5}$$

The learned attention weights for target task i reflect the correlations between other tasks and task i. And all attention weights compose the task correlation matrix $\boldsymbol{A}$.

$$\begin{aligned} \boldsymbol{A}_i &= (\boldsymbol{\alpha}_{i1}, \boldsymbol{\alpha}_{i2}, \ldots, \boldsymbol{\alpha}_{iK}),\ i = 1, 2, \ldots, K, \\ \boldsymbol{A} &= (\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_K)^T. \end{aligned} \tag{6}$$

The task-specific representations for task i is $\boldsymbol{s}_i$, the combination of all task latent representations with their attention weights for task i, as follows:

$$\begin{aligned} \boldsymbol{s}_i &= \boldsymbol{A}_i (\boldsymbol{W}_1 \boldsymbol{X}_1, \boldsymbol{W}_2 \boldsymbol{X}_2, \ldots, \boldsymbol{W}_K \boldsymbol{X}_K)^T \\ &= \sum_{j=1}^{K} \boldsymbol{\alpha}_{ij} \boldsymbol{W}_j \boldsymbol{X}_j,\ i = 1, 2, \ldots, K. \end{aligned} \tag{7}$$

We perform multi-head attention mechanism on TRAN, which allows H independent attention networks to learn the attention weights in parallel and applies linear transformation $\boldsymbol{W}_H = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_H)$ to combine them. The final task-specific representation for task i is processed as follows:

$$\begin{aligned} \boldsymbol{s}_i &= \boldsymbol{W}_H (\boldsymbol{A}_i^1, \ldots, \boldsymbol{A}_i^H)^T (\boldsymbol{W}_1 \boldsymbol{X}_1, \ldots, \boldsymbol{W}_K \boldsymbol{X}_K)^T \\ &= \sum_{h=1}^{H} \sum_{j=1}^{K} \boldsymbol{w}_h \boldsymbol{\alpha}_{ij}^h \boldsymbol{W}_j \boldsymbol{X}_j,\ i = 1, 2, \ldots, K. \end{aligned} \tag{8}$$

The illustration of feature learning is presented in Fig. 1.

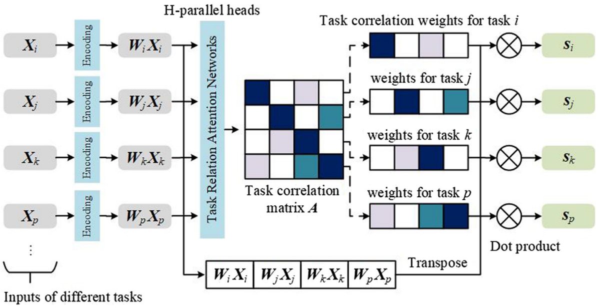


Fig. 1. Illustration of the feature learning with Task Relation Attention Networks.

3.3 Prediction Layer

For multi-task prediction, each task is equipped with a feed-forward sub-layer to convert the final task-specific representations to the predicted values. Each feed-forward sub-layer consists of two layers: the first one $\boldsymbol{W}_i^e$ is a fully-connected neural network with ReLU activation and skip-connection for embedding the final representations; the second one $\boldsymbol{W}_i^p$ is a linear transformation for prediction. The formal equation is described as:

- For regression tasks,

$$\hat{\boldsymbol{Y}}_i = \boldsymbol{W}_i^p \left(\boldsymbol{W}_i^e \boldsymbol{s}_i + \boldsymbol{s}_i \right). \tag{9}$$

- For classification tasks,

$$\hat{\boldsymbol{Y}}_i = \text{softmax} \left(\boldsymbol{W}_i^p \left(\boldsymbol{W}_i^e \boldsymbol{s}_i + \boldsymbol{s}_i \right) \right). \tag{10}$$

3.4 Objective Function

For multi-task learning, all tasks are jointly trained by optimizing a joint loss function L_{joint}. Given the inputs $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_K\}$ and labels $\boldsymbol{Y} = \{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_K\}$, the joint loss function is defined as:

$$L_{joint}(\boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{K} \lambda_i L_i(\boldsymbol{X}_i, \boldsymbol{Y}_i) + \beta_W ||\boldsymbol{W}||_F^2 + \beta_A ||\boldsymbol{A} - \boldsymbol{I}||_F^2, \tag{11}$$

where the first item is the combination of the task-specific losses L_j with their loss weights λ_j; the second item is the regularization for all the trainable parameters $\boldsymbol{W}$; the third item is the regularization for the learned attention correlation matrix $\boldsymbol{A}$ to ensure the auto-correlations of tasks. For each task, the task-specific loss is defined as:

- Mean squared error (MSE) for regression tasks,

$$L_i(\boldsymbol{X}_i, \boldsymbol{Y}_i) = \text{MSE}(\hat{\boldsymbol{Y}}_i, \boldsymbol{Y}_i). \tag{12}$$

- Cross entropy for classification tasks,

$$L_i(\boldsymbol{X}_i, \boldsymbol{Y}_i) = \text{CrossEntropy}(\hat{\boldsymbol{Y}}_i, \boldsymbol{Y}_i). \tag{13}$$

4 Experiments

4.1 Dataset

The performance of the proposed method is evaluated on three datasets: Census-income dataset, FashionMnist dataset and Sarcos dataset.

- **Census-income dataset**: The Census-income dataset is from UCI Machine Learning Repository [2]. It is extracted from the 1994 Census database, which contains 299,285 instances of demographic information for American adults. We construct two multi-task learning problems based on 40 features.
  - Task 1: predict whether the income exceeds $50K;
    Task 2: predict whether this person is never married.
  - Task 1: predict whether the education level of this person is at least college;
    Task 2: predict whether this person is never married.

- **FashionMnist dataset**: The samples in FashionMnist are 28 × 28 grayscale images with 10-class labels [19], similar to Mnist. We construct a multi-task learning problem: Task 1 is the original 10-class classification task; Task 2 is predict if the objects are shoes, or female products, or another type. All task shares the same inputs.
- **Sarcos dataset**: This is a regression dataset [4] where the goal is to predict the torque measured at each joint of a 7 degrees-of-freedom robotic arm, given the current state, velocity, and acceleration measured at each joint (7 torques for 21-dimensional inputs). Following the procedure of [4], we have 7 regression tasks, where each task is to predict one torque.

4.2 Baselines

The baseline methods to be compared with are as follows:

- **LASSO**: This is the classic linear method, learning each task independently with L1-norm regularization.
- **Bayesian**: Another linear method, learning each task independently with Bayesian inference.
- **Neural Networks (NN)**: For regression tasks and general classification tasks, we apply fully-connected neural networks with one hidden layer. For image classification, we apply single-layer convolutional neural networks.
- **Shared-bottom MTL**: This is a typical MTL method, where all tasks share the bottom hidden layers and have top sub-layers for prediction. In this method, the task relationships are predefined and fixed.
- **L2-Constrained MTL**: This is a classical MTL method [6], where the parameters of different tasks are shared softly by an L2-constraint. Given two tasks, the prediction of each task can be described as:

$$\hat{\boldsymbol{Y}}_1 = f(\boldsymbol{X}_1, \theta_1),\ \hat{\boldsymbol{Y}}_2 = f(\boldsymbol{X}_2, \theta_2), \tag{14}$$

  where θ_1, θ_2 are the parameters of each task. And the objective function of multi-task learning is:

$$L_1\left(\boldsymbol{Y}_1, \hat{\boldsymbol{Y}}_1\right) + L_2\left(\boldsymbol{Y}_2, \hat{\boldsymbol{Y}}_2\right) + \alpha||\theta_1 - \theta_2||_2^2, \tag{15}$$

  where α is a hyper-parameter. This method models the task relationships with the magnitude of α.
- **Cross-stitch Networks (CSN)**: This is a deep learning based MTL method [14]. The knowledge is shared between tasks by a linear units, call cross-stitch. Given two tasks, $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ are the outputs of previous hidden layers of each task, and the outputs of cross-stitch are described as:

$$\begin{bmatrix} \tilde{\boldsymbol{h}}_1 \\ \tilde{\boldsymbol{h}}_2 \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \end{bmatrix}, \tag{16}$$

  where $\alpha_{ij},\ i, j = 1, 2$ are trainable linear parameters representing the knowledge transfer.

- **Multi-gate Mixture-of-Experts (MMoE)**: It adopts the multi-gate mixture-of-experts structure to MTL [13]. In this method, there is a group of expert networks as the bottom layers, and the top task-specific layers can utilize the experts differently with gating mechanism.
- **Multiple Relational Attention Networks (MRAN)**: It was recently proposed [21], applying attention networks to model multiple types of relationships in MTL. However, the task relationships in this method are statically modeled. The relationships are determined in the training phase, and in the testing phase, the relationships are fixed. Our method is able to dynamically capture the task relationships, and we will discuss about the differences between this method and ours.

4.3 Performance Comparison

The proposed method is Task Relation Attention Networks (TRAN), and MH-TRAN means it is equipped with multi-head mechanism. Note that, because the code and some datasets of baseline MRAN are not released yet, we only compare its performance in the available Sarcos dataset, which is reported in their paper.

Overall Comparison. The performance comparison on Census-income, FashionMnist and Sarcos datasets are presented in Table 1, 2 and 3 respectively. From all the tables, we can observe that both TRAN and MH-TRAN outperform other methods on all tasks, and MH-TRAN outperforms TRAN on most tasks. The average relative improvements of MH-TRAN in all tasks are 4.68% (L2-Norm), 5.78% (CSN) and 2.45% (MMoE) for Census-income dataset, 2.94% (L2-Norm), 1.68% (CSN) and 2.85% (MMoE) for FashionMnist dataset and 67.95% (L2-Norm), 55.09% (CSN), 40.50% (MMoE) and 25.29% (MRAN) for Sarcos dataset.

Comparison Between Different MTL Methods. Compare with TRAN, both CSN and MMoE are based on linear models, and MMoE model task relationships implicitly, while our method is based on attention networks to explicitly model task relationships. The fact that TRAN outperforms CSN and MMoE, indicating the advantages of our method. Besides, compared with statically modeling task relationships (MRAN), TRAN is able to dynamically control the knowledge transfer, and the fact that TRAN outperforms MRAN demonstrates the effectiveness of this key component. Although MRAN includes different types of relationships, our method still outperforms it with the relative improvements of 25.29%.

4.4 Analysis for Key Components

Task Relationships and Knowledge Transfer. We illustrate the task correlations learned by TRAN in Fig. 2.

In overall, all tasks are strongly correlated to themselves. And for different target tasks, the contributions of the other tasks vary a lot, e.g., the relationships in Sarcos dataset in Fig. 2(c). We can observe the differences between traditional methods and TRAN. In traditional methods, the task correlations are predefined and equal for each task, however, TRAN captures their differences. In Sarcos dataset, for task 7, the contribution of task 1 is apparently less than the others. If the method relies on the pre-definition of equal task correlations, there may exist negative knowledge transfer hurting the performance. From the performance comparison, we can observe that TRAN outperforms the traditional methods with pre-definition, which demonstrates the effectiveness of adaptively capturing the task correlations.

Table 1. Performance comparison on the Census-income dataset in terms of AUC.

Method	AUC/Census-income dataset			
	Income	Marital	Education	Marital
LASSO	0.8849	0.9367	0.7695	0.9367
Bayesian	0.9267	0.8604	0.8209	0.8718
NN	0.8904	0.9387	0.8179	0.8841
Shared-bottom	0.8997	0.9379	0.8201	0.8973
L2-Norm	0.8967	0.9398	0.8174	0.9366
CSN	0.8913	0.9401	0.8219	0.8996
MMoE	0.9298	0.9523	0.8444	0.9422
TRAN	0.9412	0.9693	0.8566	**0.9792**
MH-TRAN	**0.9501**	**0.9721**	**0.8613**	0.9789

Table 2. Performance comparison on the FashionMnist dataset in terms of accuracy.

Method	Accuracy/FashionMnist	
	10-class	3-class
LASSO	0.8691	0.8879
Bayesian	0.8698	0.8981
NN	0.9032	0.9103
Shared-bottom	0.9044	0.9182
L2-Norm	0.9031	0.9173
CSN	0.9107	0.9324
MMoE	0.9079	0.9142
TRAN	0.9180	0.9497
MH-TRAN	**0.9207**	**0.9533**

For Census-income dataset, we have two multi-task learning problems, and marital task appears in both group I and II accompanied with different tasks. As the performance comparison in Table 1, marital task of TRAN in group II performs better than the one in group I. And from the illustration, we can observe that the task correlations in group II are stronger than the correlations in group I. This indicates there are more positive knowledge transfer in group II, which contributes to the improved performance.

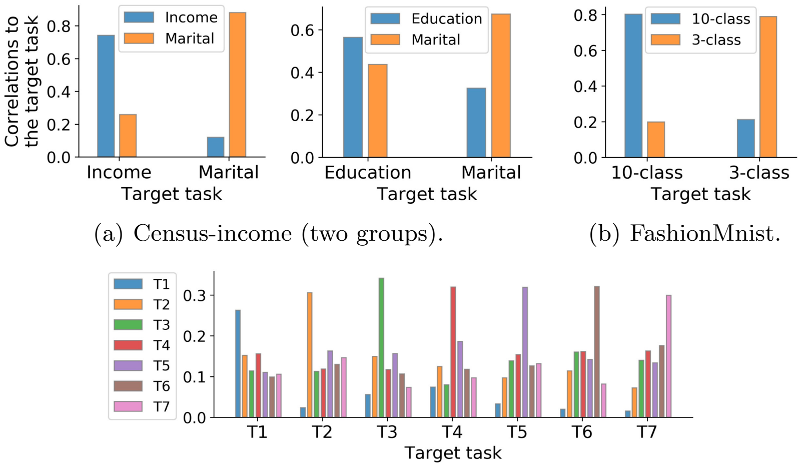
In order to verify our observation, we assess the practical strengths of task relationships in group I and II, because in general, stronger task correlations imply that there are more positive knowledge transfer. According to the works of [13], the Pearson correlations of the labels of different tasks can be used as the quantitative indicator of task relationships, because the Pearson correlations of labels are positively correlated to the strength of task relationships. The Pearson correlation in group I is 0.1784, and the one in group II is 0.2396. This indicates that there is supposed to be more positive knowledge transfer in group II, corresponding to our observation. This demonstrates that TRAN does capture the practical task correlations and control the positive knowledge transfer to help improve the performance.

Dynamically Control the Knowledge Transfer. The task relationships are not fixed, but vary slightly from different samples. We aim to dynamically capture the task relationships from different samples using TRAN.

We randomly select 8 samples from the testing samples of Sacros dataset, and provide an illustration of their dynamically learned task correlations in Fig. 3. From the correlations, we can observe that there is a slight variety in the task relationships in different samples. This demonstrate that TRAN does capture the dynamic task relationships, and the performance comparison in Table 1, 2 and 3 indicates TRAN controls the knowledge transfer to improve the performance using the dynamically learned relationships.

Table 3. Performance comparison on the Sarcos dataset in terms of RMSE.

Method	RMSE/Sarcos dataset						
	T_1	T_2	T_3	T_4	T_5	T_6	T_7
LASSO	5.848	5.159	3.153	3.501	0.410	0.952	0.733
Bayesian	5.576	4.763	3.014	3.119	0.366	0.910	0.661
NN	5.534	4.936	3.002	3.375	0.393	0.919	0.704
S-bottom	4.476	4.396	2.172	3.184	0.396	0.873	0.724
L2-Norm	4.773	4.507	2.255	3.254	0.382	1.110	0.721
CSN	3.774	3.456	1.749	1.885	0.315	0.553	0.403
MMoE	3.094	2.329	1.328	1.335	0.284	0.431	0.358
MRAN	2.879	1.895	1.041	0.829	0.154	0.261	0.236
TRAN	1.955	1.388	0.833	0.798	0.132	**0.242**	0.214
MH-TRAN	**1.937**	**1.371**	**0.793**	**0.771**	**0.122**	0.245	**0.212**



(a) Census-income (two groups).

(b) FashionMnist.

(c) Sarcos dataset.

Fig. 2. Illustration of the task correlations learned by TRAN.

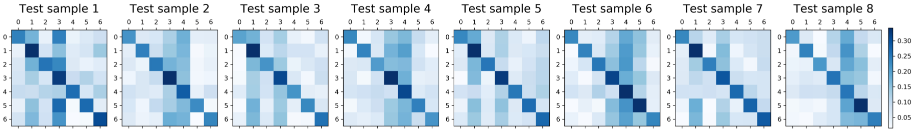


Fig. 3. Illustration of the dynamic task relationships on the Sarcos dataset. We randomly select 8 samples from the testing dataset and visualize their attention correlation matrices.

5 Conclusion

In this paper, we propose Task Relation Attention Networks to adaptively capture the task relationships, replacing the pre-defined ones in traditional MTL methods. Based on the learned relationships, the positive and negative knowledge transfer can be dynamically balanced in different samples. As a result, a better task-specific representation is obtained and leads to improved performance. In addition, the learned correlation matrix presents the dynamic transfer pattern, making the MTL method more explicable. To evaluate its performance, we conduct experiments on various datasets, including regression and classification tasks. Both classical and state-of-the-art MTL methods are employed to provide benchmarks. The experimental results and analyses demonstrate the effectiveness of our method, and its advantages over other methods.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (GrantNos.: 61673025, 61375119), and the National Key R&D Program of China (Grant Nos.: 2018AAA0100300, 2018AAA0102301), and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China (Grant No. 2015CB352302).

References

1. Allamanis, M., Peng, H., Sutton, C.: A convolutional attention network for extreme summarization of source code. In: International Conference on Machine Learning, pp. 2091–2100 (2016)
2. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
3. Caruana, R.: Multitask learning. Mach. Learn. **28**(1), 41–75 (1997)
4. Ciliberto, C., Rudi, A., Rosasco, L., Pontil, M.: Consistent multitask learning with nonlinear output relations. In: Advances in Neural Information Processing Systems, pp. 1986–1996 (2017)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Duong, L., Cohn, T., Bird, S., Cook, P.: Low resource dependency parsing: cross-lingual parameter sharing in a neural network parser. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol. 2: Short Papers. vol. 2, pp. 845–850 (2015)
7. Hashimoto, K., Xiong, C., Tsuruoka, Y., Socher, R.: A joint many-task model: growing a neural network for multiple NLP tasks. arXiv preprint arXiv:1611.01587 (2016)
8. Kaiser, L., et al.: One model to learn them all. arXiv preprint arXiv:1706.05137 (2017)
9. Li, Y., Fu, K., Wang, Z., Shahabi, C., Ye, J., Liu, Y.: Multi-task representation learning for travel time estimation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1695–1704. ACM (2018)
10. Lohit, S., Wang, Q., Turaga, P.: Temporal transformer networks: joint learning of invariant and discriminative time warping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 12426–12435 (2019)
11. Lu, Y., Kumar, A., Zhai, S., Cheng, Y., Javidi, T., Feris, R.: Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5334–5343 (2017)
12. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015)
13. Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., Chi, E.H.: Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1930–1939. ACM (2018)
14. Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-stitch networks for multi-task learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3994–4003 (2016)

15. Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 (2017)
16. Torrey, L., Shavlik, J.: Transfer learning. In: Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques, pp. 242–264. IGI Global (2010)
17. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
18. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 vol. 1, no. 2 (2017)
19. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
20. Yang, Y., Hospedales, T.: Deep multi-task representation learning: a tensor factorisation approach. arXiv preprint arXiv:1605.06391 (2016)
21. Zhao, J., Du, B., Sun, L., Zhuang, F., Lv, W., Xiong, H.: Multiple relational attention network for multi-task learning. In: Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E., Karypis, G. (eds.) Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, 4–8 August 2019, pp. 1123–1131. ACM (2019). https://doi.org/10.1145/3292500.3330861

RPP Algorithm: A Method for Discovering Interesting Rare Itemsets

Sadeq Darrab(✉), David Broneske, and Gunter Saake

University of Magdeburg, Magdeburg, Germany
{sadeq.darrab,david.broneske,gunter.saake}@ovgu.de

Abstract. The importance of rare itemset mining stems from its ability to discover unseen knowledge from datasets in real-life domains, such as identifying network failures, or suspicious behavior. There are significant efforts proposed to extract rare itemsets. The RP-growth algorithm outperforms previous methods proposed for generating rare itemsets. However, the performance of the RP-growth degrades on sparse datasets, and it is costly in terms of time and memory consumption. Hence, in this paper, we propose the RPP algorithm to extract rare itemsets. The advantage of the RPP algorithm is that it avoids time for generating useless candidate itemsets by omitting conditional trees as RP-growth does. Furthermore, our RPP algorithm uses a novel data structure, RN-list, for creating rare itemsets. To evaluate the performance of the proposed method, we conduct extensive experiments on sparse and dense datasets. The results show that the RPP algorithm is around an order of magnitude better than the RP-growth algorithm.

Keywords: Rare itemset · RN-list · RPP

1 Introduction

Since the emergence of data mining, there is a plethora of methods introduced to extract frequent itemsets [2]. However, rare itemset mining (RIM) discovers unusual events (known as rare itemsets). Discovering rare itemsets provides useful information in many domains since it extracts an uncommon valuable knowledge from datasets. RIM is widely used in many applications such as market basket analysis [2], predicting telecommunication equipment failures [8], identifying fraudulent credit card transactions [9], medical diagnosis [10], and adverse drug reactions [15]. For instance, in the health care domain, frequently discovering (known) complications are less interesting than rarely (unexpected) complications that may be more important to be discovered as early as possible. Therefore, discovering unusual events (rare itemsets) is more interesting, and it comes into the focus since it may help us to avoid adverse consequences. Traditional mining methods extract rare itemsets by setting a very low support threshold, which leads to an over-generation of itemsets. Thus, analyzing a large

Y. Tan et al. (Eds.): DMBD 2020, CCIS 1234, pp. 14–25, 2020.
https://doi.org/10.1007/978-981-15-7205-0_2

amount of itemsets is computationally expensive. To overcome this issue, a high support threshold is used, which leads to a loss of interesting rare itemsets. Thus, extracting rare itemsets is a challenge, and it is called a rare item problem [11]. To handle the rare item problem, there are various methods proposed [1,3,4,6]. These methods can be classified based on their traversal of the search space into breadth-first search and depth-first search [19]. Breadth-first search methods use the most well-known algorithm called an apriori algorithm [2] to mine rare itemsets. These methods inherit apriori algorithm's drawbacks (overly huge candidate sets, redundant scans over the data). To overcome these shortcomings, depth-first search methods have been introduced to mine rare itemsets by utilizing an FP-tree structure [12]. A prominent FP-tree-based algorithm to mine rare itemsets is RP-growth [6]. The RP-growth algorithm solves the shortcomings of apriori-based algorithms in terms of time and memory consumption. However, it generates an unnecessary amount of conditional trees, which compromises search time and storage consumption in sparse datasets [14]. However, especially sparse datasets contain many uncommon patterns. Hence, how to design an efficient mining method for rare itemsets mining on sparse datasets is still critical research. In this paper, we propose the RPP algorithm for discovering rare itemsets. The RPP algorithm is inspired by a novel data structure, N-list [13], to generate interesting rare itemsets. The following steps summarize the contribution of this paper.

- An RPPC-tree is constructed to maintain all information needed to extract rare itemsets. It is constructed from transactions that contain at least one rare item.
- An RN-list of all interesting rare items is created.
- The RN-list of rare items are used in the mining process to generate the whole set of rare itemsets by intersecting these lists.

The rest of the paper is organized as follows. The concept of rare itemset mining is presented in Sect. 2 and relevant literature is introduced in Sect. 3. The details of the proposed algorithm, the RPP algorithm, is explained in Sect. 4. Sections 5 and 6 present the performance results and conclusion, respectively.

2 Preliminaries

To understand the basic concepts of RIM, let us consider the following motivating example.

Motivating example: given a transaction dataset DB in Table 1, let maximum support threshold ($maxSup$) and rare support threshold ($minSup$) be 0.80, 0.40, respectively. The task of rare itemset mining is to extract the set of all rare itemsets with support not less than $minSup$ but also not exceeding $maxSup$.

Definition 1. Rare itemsets: An itemset X is called rare itemset if its occurrence frequency in a given dataset is less than the user given minimum support threshold, $freqSup$, such that $Sup(X) < freqSup$.

Definition 2. Interesting rare itemsets: An itemset X whose support satisfies the following conditions is called the interesting rare itemset: $Sup(X) < maxSup \wedge Sup(X) \geq minSup$.

For instance, the itemset {ce : 2} in the motivating example is called interesting rare itemset since $Sup(X) = 0.40$ which is less than $maxSup$ and it is greater or equals to the $minSup$.

Table 1. A simple dataset.

TID	Items	Ordered items
1	a, b, c, d	b, c, a, d
2	b, d	b, d
3	a, b, c, e	b, c, a, e
4	c, d, e, h	c, d, e
5	a, b, c, g	b, c, a

3 Related Work

Rare itemset methods can be classified based on the exploration of the search space into two categories. The first category includes level-wise exploration methods that depend on the downward closure property for pruning uninteresting itemsets [1,2]. The second category uses a level-depth exploration [6,21].

3.1 Level-Wise Exploration Methods

Level-wise exploration methods work similar to an apriori algorithm [2]. These methods generate k-itemsets (itemsets of cardinality k) with using $(k-1)$-itemsets by utilizing the downward closure property. In [1], an apriori-inverse algorithm is proposed to extract rare itemsets that have a support below a maximum support threshold ($maxSup$). It works similar to the apriori algorithm except that in the first step the apriori-inverse algorithm generates 1-itemsets that do not satisfy $maxSup$. In [3], two algorithms are presented to detect rare itemsets. In the first step, an apriori-rare algorithm identifies the minimal rare itemsets (itemsets that do not have any subset which is rare). In the second step, the minimal rare itemsets are used by an MRG-Exp algorithm as a seed for generating the whole set of rare itemsets. A method in [4] stated that exploring the itemset space in a bottom-up fashion is costly in term of time and memory consumption since it is common that rare itemsets are found at the top of the search space (i.e., in the last iteration steps). Instead, they have proposed AfRIM using a top-down traversal approach. AfRIM starts with the largest n-itemset that contains all unique items found in a dataset. It discovers the candidate n-1-itemset subsets from rare n-itemsets and collects the interesting rare itemsets. Still, AfRIM is a time-consuming method since it considers zero-itemsets in the mining process as it begins from the largest itemset that contains all distinct items in the dataset. To avoid zero-itemset generation, the rarity algorithm [5] is proposed for retrieving rare itemset. It works similar to AfRIM except that it starts from items in the longest transaction in the dataset.

In [11], the first breadth-first search algorithm MSapriori, is proposed to extract both frequent and rare itemsets under different support thresholds (MIS). The MSapriori algorithm and its optimizations [16–18] extract frequent itemsets including rare ones by assigning lower support thresholds for rare (infrequent) itemsets than for most common (frequent) itemsets. Hence, they work similarly to the former apriori [2] with the following major difference: It declares an itemset (frequent and rare) as an interesting itemset if its support satisfies the lowest MIS of items within it.

3.2 Level-Depth Exploration Methods

The above methods use the formal apriori algorithm [1], which is computationally expensive since 1) they employ candidate generation and test fashion, and 2) redundant scans over the dataset for each new generated candidate itemset. Furthermore, these methods spend a huge amount of time searching for the whole candidate itemsets (including useless itemsets whose support is zero) in order to identify the rare itemsets. To overcome these problems, several methods [21–24] are presented to discover most common (frequent) itemsets including unusual (rare itemsets). CFP-growth [23] and its optimization CFP-growth++ [24] algorithms scan the dataset once to build a CFP-tree. Then, they reconstruct the tree by employing several punning and merging techniques. Reconstruction of the tree is computationally expensive in terms of memory and time consumption. To overcome this shortcoming, the MISFP-growth algorithm [21] is proposed to extract frequent itemsets including rare ones under MIS. The MISFP-growth efficiently builds the tree without a need for the reconstruction phase. Unlike the FP-growth based methods, mis-eclat [22] utilizes a vertical representation of data to extract both frequent and rare itemsets. Although these methods address the rare itemset problem, they suffer from overly huge itemsets since they extract both frequent and rare itemsets.

Focusing only on rare itemsets, the RP-Tree algorithm [6] is proposed to extract rare itemsets without the expensive itemset generation and pruning steps. It utilizes a tree data structure, the FP-Tree [12]. The RP-Tree is based on FP-growth, which uses a divide-and-conquer approach to generate rare itemsets. However, for each rare item, RP-growth generates a conditional tree in the mining process for generating rare itemsets. The RP-growth method becomes costly when datasets are sparse since building RP-trees, which bases on conditional pattern, recurrently makes RP-growth inefficient. It is common that rare itemsets mostly occur in sparse datasets. Hence, how to design efficient mining methods for mining rare itemsets is still a critical research problem.

4 Mining Rare Itemsets with Rare Pre Post (RPP) Algorithm

Inspired by PrePost algorithm in [13], we propose the rare pre post algorithm (RPP) to extract meaningful rare itemsets. The RPP involves three sequential phases as follows.

1. At the first step, we build a Rare Pre-Post Code tree (RPPC-tree) to store all information needed to extract rare itemsets. The RPPC-tree is constructed

by adding transactions that contain at least one rare item. Each node in the RPPC-tree consists of four fields: item name, count, pre-order rank and post-order rank. Item name represents the name of the item that this node represents. Count registers the number of transactions that reaches this node. Pre-order and post-order ranks stand for the number of traversed nodes from the root to this node when using pre or post order traversal.

2. At the second step, we traverse the RPPC-tree to generate the rare pre post code (RPP-code) for each node in the RPPC-tree. For a node X in the RPPC-tree, the tuple of {`(X-pre-order, X-post-order): count`} is called the RPP-code of X. For performance reasons, the RPP-codes are sorted in an ascending order according to their pre-order values. The RPP-codes of nodes that contain the same item X in the RPPC-tree is called RN-list of X, which is generated in Step 3.
3. For the mining process, the RPP algorithm constructs RN-lists by considering only the rare items. The RPP method iteratively constructs RN-lists of rare itemsets of length k by intersecting RN-lists of rare itemsets of length k−1. Finally, this process is terminated when there is no more RN-list that can be intersected with.

To show how the RPP method works, let us consider our motivating example to illustrate the three required phases of RPP algorithm in the following.

4.1 Construction of the RPPC-tree

To hold the necessary information from a dataset, the RPPC-tree is constructed. The tree is built by scanning the dataset twice. In the first scan, the support of 1-itemsets is counted. In the second scan, we add a transaction into the tree if the transaction has at least one rare item (item with support no greater than *maxSup* and no less than *minSup*). The RPPC-tree looks like the RP-tree and it can be defined as follows.

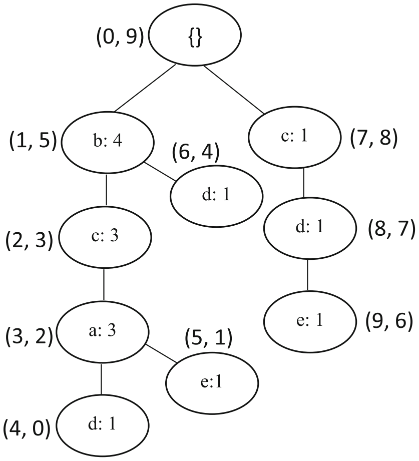


Fig. 1. RPPC-tree after adding all transactions in Table 1.

Definition 3. The RPPC-tree is a tree structure with the following properties: It consists of one root node, which is labeled as “null” , and a set of nodes as the children of the root. Except the root, each node in the RPPC-tree consists of four fields: item-name, count, pre-order, and post-order. Item name holds the name of the item that this node represents. Count registers the number of transactions that reaches this node. A pre-post rank of a node registers the pre-order and post-order for each node within the tree, i.e., its position when traversing the tree in pre-order or post-order fashion, respectively.

Following the motivating example, we show how the RPPC-tree is constructed. To build the RPPC-tree, the dataset is scanned twice. In the first scan, we calculate the support of the 1-items and eliminate useless items with support less than *minSup*. For instance, the items g, h are discarded since their support equals 1 which is less than $minSup = 2$. The interesting items are sorted in descending support order as in the right column in Table 1. In the second scan, the transactions in the right column in the Table 1 are used to construct the RPPC-tree. Figure 1 shows the final RPPC-tree after adding all transactions. As it can be seen from Fig. 1, each node registers the item that this node represents, the number of transactions that reach this node, and the pre-post rank of the node. For instance, the node c, its item name is c, its count = 3 and (2, 3) is a pre-post rank of the node c.

4.2 Generating RN-list of Items

The main purpose of the RPPC-tree is to generate the RN-lists of items. The RN-lists hold all necessary information for discovering the whole set of rare itemsets. To construct the RN-lists of items, first, the pre-post codes are generated for each node in the RPPC-tree by traversing the tree in pre and post order.

Table 2. RN-lists of interesting rare items.

Item	RPP-codes	Support
b	{(1, 5): 4}	4
c	{(2, 3):3, (7, 8):1}	4
a	{(3, 2):3}	3
d	{(4, 0):1, (6, 4):1, (8, 7):1}	3
e	{(5, 1):1, (9, 6):1}	2

For a node X, its pre-post code is called RPP-code ((X.pre-order, X.post-order): count). For instance, the RPP-code of node a is ((3, 2): 3) indicating that a.pre-order is 3, a.post-order is 2, and its count is 3. The list of the RPP-codes of nodes that hold the same item X in the RPPC-tree is called the RN-list of X. The RPP-codes are sorted in ascending order of their pre-order values. For instance, a node (e: 2) in Fig. 1 has two RPP-codes which are ((5, 1):1), and ((9, 6):1). Then, the RN-list of the itemset e is {((5, 1):1, ((9, 6):1)}. The advantage of the RN-list approach is its easy support calculation. For the RN-list of itemset X, which is denoted by (x1, y1): z1, (x2, y2): z2, . . . , (xm, ym) : zm, its support is $z1 + z2 + \ldots + zm$. For instance, the RN-list of the itemset e are {((5, 1):1, ((9, 6):1)} and its support is 2. Following the motivating example, Table 2 shows the RN-list of all interesting 1-itemset.

4.3 Generation of Rare Itemsets

The rare itemsets can be generated by using the information that is contained in the RN-lists of itemsets. For RN-lists X, Y, the itemset XY can be generated if X is an ancestor of Y. We call a node X as an ancestor of node Y when: X.pre-order <Y.pre-order and X.post-order> Y.post-order (i.e., we call it an ancestor-descendant relation). To generate the itemset XY, we traverse the RPP-codes of X and compare them with the RPP-codes of Y. Then, if the ancestor-descendant relation of X and Y holds (i.e., they are in the same path), we add the RPP-code of X to the RN-list of XY. For the support count, we add Y.count to the generated RN-list of XY since the items are sorted according to their descending support (i.e., the itemset X occurs together with itemset Y at most Y.count). Following our motivating example, we illustrate this phase as follows. Table 2 contains the RN-list of all 1-itemsets. To shrink the search space, we compare the RN-list of itemsets with only the RN-list of rare items that satisfy both of $minSup = 2$ and $maxSup = 4$. Thus, the RN-lists of {a, d, e} are used to generate the whole set of rare itemsets. For instance, to generate rare itemset {ce}, we first check whether the support of item e satisfies Definition 3. We find out that $minSup = 2 \leq Sup(e) = 2 < maxSup = 4$. Then, we compare the RN-list of c with RN-list of e as follows.

1. The RN-list of the itemset c is {(2, 3): 3, (7, 8): 1}, and the RN-list of the itemset e is {(5, 1):1, (9, 6):1}. We compare each RPP-code of c with all RPP-codes of the itemset e to generate the itemset ce.
2. The RPP-code of c ((2, 3): 3) is compared with the first RPP-code of e ((5, 1):1). We notice that $2 < 5$ and $3 > 1$, which satisfies the ancestor-descendant relation. Then, we add the RPP-code {(2, 3): 1} to the RN-list of ce {(2, 3): 1}. Notice, we add the count of itemset e since it is descendant of c and cannot occur together more than e.count.
3. The RPP-code of c ((2, 3): 3) is compared with the next RPP-code of e ((9, 6):1). We find that $2 < 9$ and $3 < 6$, which does not satisfy the ancestor-descendant relation. Then, we would go for the next RPP-code of e. Since there is no further RPP-code of e, we traverse the next RPP-code of c.
4. The RPP-code of c ((7, 8): 1) is compared with the first RPP-code of e ((5, 1):1). We find that $7 > 5$ and $8 > 1$, which does not satisfy the ancestor-descendant relation. Hence, we do not add this RPP-code to the RN-list of itemset ce.
5. The RPP-code of c ((7, 8): 1) is compared with the next RPP-code of e ((9, 6):1). The ancestor-descendant relation $7 < 9$ and $8 > 1$, holds. Thus, we add ((7, 8): 1) to the RN-list of ce {(2, 3):1, (7, 8): 1}.
6. The resulted RN-list of the itemset ce is {(2, 3): 1, (7, 8): 1}.
7. The support of the itemset ce is 2. Thus, the itemset ce is interesting rare itemset since $minSup = 2 \leq sup(ce) = 2 < maxSup = 4$.

Similar to the above steps, the process is repeated for the remaining itemsets. The final rare itemsets for our example are {a: 3, e: 2, d: 2, ba:3, bd: 2, ca:3, cd: 2, ce: 2, bca: 3}. We generate rare itemsets only from rare items {a, d, e}. For instance, the RN-list of the itemset c will not be compared with the RN-list of b since the support of the itemset b is not less than the $maxSup = 4$ value.

5 Experimental Results

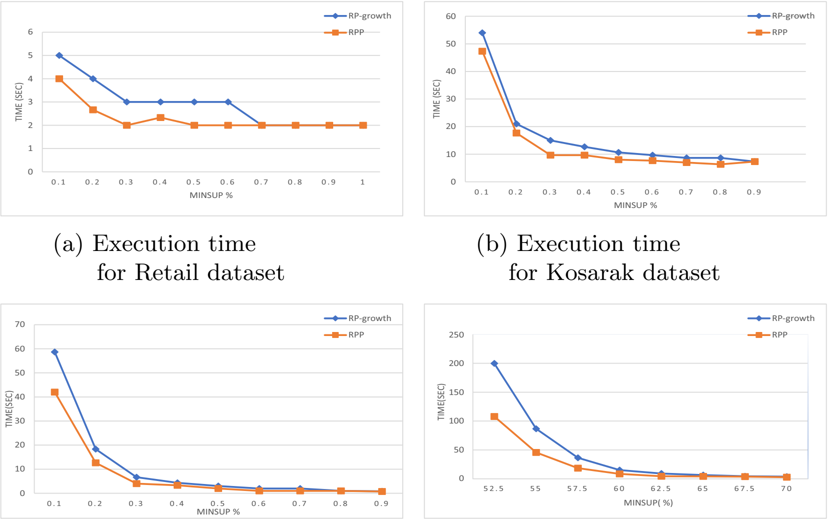
To measure the performance of the RRP algorithm, we compare its performance with the state of art algorithm for mining rare itemset, RP-growth [6]. We carried out several experiments on four real-world datasets: Mushroom, Retail, Pumsb, and Kosarak. Both, sparse datasets (Kosarak, Retail), and dense datasets (Mushroom, Pumsb) are used for the evaluation process. The characteristics of the datasets are summarized in Table 3. For each dataset, the number of transactions, the number of distinct items, and an average transaction are denoted by # of Trans, # of items, and AvgTrans, respectively. The last column in Table 3 shows the density of the datasets. The datasets are downloaded from FIMI [7]. The experiments run on windows 10, 64 bit operating system, Intel Core i7-7700HQ CPU 2.80 GHz with 16 GB main memory, and 1 TB hard disk. The algorithms are implemented in Java to have a common implementation environment. The source code of RP-growth is downloaded from [20].

Table 3. The characteristics of the datasets.

Dataset	Size (MB)	# Items	# Trans	AvgTrans	MaxSup	MinSup (%)
Mushroom	19.3	119	8124	23	0.01	{0.1, 0.2,. . . , 0.9}
Retail	4.2	16,470	88,126	10.3	0.1	{0.1, 0.2,. . . , 1}
Pumsb	16.3	2,113	49046	74	0.8	{52.5, 55, . . . , 70}
Kosarak	30.5	41,271	990,002	8.1	0.01	{0.1, 0.2,. . . , 0.9}

5.1 Execution Time

We compare our RPP algorithm with RP-growth to evaluate the execution time on all the datasets in Table 3. For all experiments, we use two support thresholds ($maxSup$ and $minSup$) to extract rare itemsets. For each experiment, we fix the $maxSup$ support threshold and vary the minimum rare support threshold ($minSup$), as shown in columns ($maxSup, minSup$) in Table 3. The interesting rare itemsets should be less than $maxSup$ and greater or equal to $minSup$. In each graph, the X-axis represents the varied values of $minSup$, whereas the Y-axis stands for the execution time. Fig. 2(a)–(d) show the performance of the proposed algorithm, RPP, and RP-growth algorithm in terms of runtime. The graphs show that our RPP algorithm outperforms RP-growth for all datasets. The advantage of our RPP algorithm is that it utilizes only RN-lists of rare itemsets to generate rare itemsets. In contrast, the RP-growth is costly since it builds conditional trees for each rare item during the mining process. It can be noticed from the graphs that the RPP algorithm is orders of magnitude faster than RP-growth at low $minSup$ values. This is a significant improvement since most of the interesting rare itemsets can be generated with low $minSup$ value. It can be observed from the graphs that the performance is approximately the same when increasing the $minSup$ value. They gained the same performance when the $minSup$ value increased since the difference between $maxSup$ and $minSup$ is decreased, and a small number of rare itemsets will be generated.



(a) Execution time for Retail dataset

(b) Execution time for Kosarak dataset

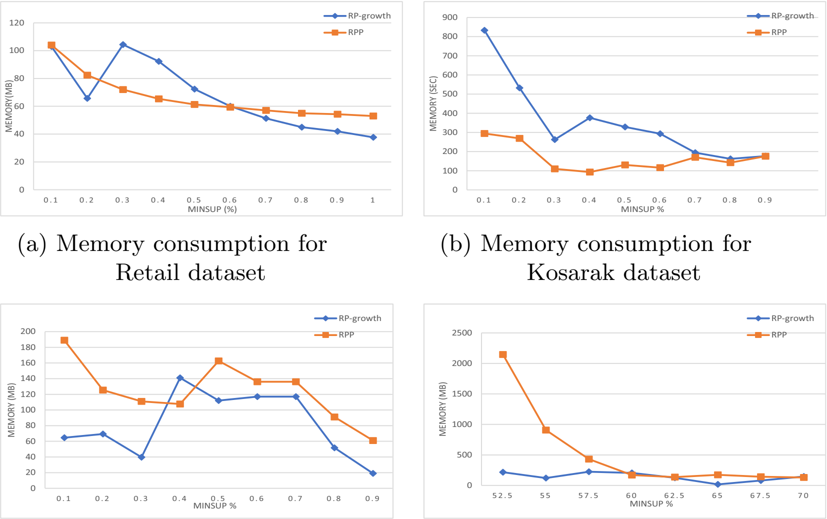
(c) Execution time for Mushroom dataset

(d) Execution time for Pumsb dataset

Fig. 2. Execution time of RP-growth and our RPP algorithm for different datasets.

5.2 Memory Consumption

To evaluate the memory cost, we use the same factors that are shown in columns ($maxSup, minSup$) in Table 3. Fig. 3(a)–(d) show the memory cost of RPP and RP-growth algorithms on all datasets at different $minSup$ values. Similar to the execution time figures, the $maxSup$ value is fixed, while the $minSup$ value is changed. In all figures, $minSup$ values are located at X-axes, whereas the Y-axes represent the memory consumption by both algorithms. Fig. 3(a)–(b) show the memory cost of RPP algorithm and RP-growth algorithm on the retail and Kosarak datasets. The graphs show that our RPP algorithm consumes less memory than the RP-growth algorithm for the sparse datasets. For the retail dataset, the RPP algorithm consumes less memory than the RP-growth when the $minSup$ value exceeds 0.3%, and it consumes a little more memory than the RP-growth when the $minSup$ value exceeds 0.6%. Fig. 3(b) shows the memory consumption of RPP and RP growth algorithms on the very sparse datasets, Kosarak. Notably, the RPP algorithm consumes a little less memory than RP-growth for all $minSup$ values. For dense datasets (Mushroom, Pumsb), Fig. 3(c) and Fig. 3(d) show the memory consumption of RPP and RP-growth algorithms. The RPP algorithm consumes more memory than the RP-growth algorithm on Mushroom and Pumsb datasets.



(a) Memory consumption for Retail dataset

(b) Memory consumption for Kosarak dataset

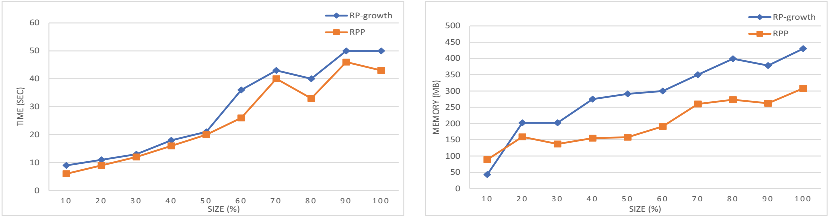
(c) Memory consumption for Mushroom dataset

(d) Memory consumption for Pumsb dataset

Fig. 3. Memory consumption of RP-growth and our RPP algorithm for different datasets.

5.3 Scalability

To evaluate the scalability of RPP algorithm and the RP-growth, we choose the largest dataset, Kosarak. It contains about 1 million transactions. The dataset is equally divided into ten parts. For each experiment, we add 10 % to the previous accumulative parts. Fig. 4(a)–(b) demonstrates the experimental results of the RPP and RP-growth algorithms in terms of time and memory consumption. The graphs illustrate that the proposed RPP algorithm scales better than RP-growth when increasing the size of the dataset. The RPP algorithm requires less time and memory since it depends on RN-lists of rare itemsets during the mining process. For RP-growth, it needs to traverse a big search space and generate a large number of conditional trees to generate rare itemsets. To sum up, our RPP algorithm is faster than RP-growth on all datasets that are given in Table 3. For memory consumption, our RPP algorithm requires less memory on sparse datasets, and it consumes more memory on dense datasets. Finally, when the size of the dataset increases, our RPP algorithm scales better than RP-growth.



(a) Scalability of execution time.

(b) Scalability of memory consumption.

Fig. 4. Scalability of our RPP algorithm compared to RP-growth for the Kosarak dataset.

6 Conclusion

In this paper, we proposed the RPP algorithm to discover the whole set of rare itemsets. The RPP algorithm utilizes RN-lists of items, which contains the whole information needed to generate rare itemsets. The RPP algorithm shrinks the search since 1) it avoids redundant scans of the dataset, 2) it extracts the whole set of interesting rare itemsets without generating conditional trees, and 3) the support of the generated itemsets is calculated by intersection operations. To test the performance of the RPP algorithm, we compared its performance with the well-known algorithm for rare itemsets mining on dense and sparse datasets, the RP-growth algorithm. The experimental results showed that the RPP algorithm is significantly better than the RP-growth algorithm in terms of execution time, memory cost, and scalability.

References

1. Koh, Y.S., Rountree, N.: Finding sporadic rules using Apriori-inverse. In: Ho, T.B., Cheung, D., Liu, H. (eds.) Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), pp. 97–106. Springer, Heidelberg (2005). https://doi.org/10.1007/11430919_13
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of 20th International Conference on Very Large Data Bases (VLDB), VLDB, pp. 487–499 (1994)
3. Szathmary L., Napoli A., Petko V.: Towards rare itemset mining. In: 19th International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, pp. 305–312 (2007)
4. Adda, M., Wu L., Feng, Y.: Rare itemset mining. In: Sixth International Conference on Machine Learning and Applications (ICMLA), IEEE, pp. 73–80 (2007)
5. Troiano, L., Scibelli, G., Birtolo, C.: A fast algorithm for mining rare itemsets. In: Proceedings of 9th International Conference on Intelligent Systems Design and Applications, IEEE Computer Society Press, pp. 1149–1155 (2009)
6. Tsang, S., Koh, Y.S., Dobbie, G.: RP-Tree: rare pattern tree mining. In: Cuzzocrea, A., Dayal, U. (eds.) Data Warehousing and Knowledge Discovery (DaWaK). Lecture Notes in Computer Science, pp. 277–288. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23544-3_21

7. Frequent Itemset Mining Dataset Repository. http://fimi.uantwerpen.be/data/
8. Bhatt, U.Y., Patel P.A.: An effective approach to mine rare items using Maximum Constraint. In: Intelligent Systems and Control (ISCO), IEEE, pp. 1–6 (2015)
9. Weiss, G.M.: Mining with rarity: a unifying framework. In: ACM SIGKDD Explorations Newsletter, pp. 7–19 (2004)
10. Szathmary, L., Valtchev, P., Napoli, A., Godin, R.: Efficient vertical mining of minimal rare itemsets. In: Proceedings of 9th International Conference Concept Lattices and Their Applications (CLA), pp. 269–280 (2012)
11. Lui, C.-L., Chung, F.-L.: Discovery of generalized association rules with multiple minimum supports. In: Zighed, D.A., Komorowski, J., Zytkow, J. (eds.) European Conference on Principles of Data Mining and Knowledge Discovery, pp. 510–515. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45372-5_59
12. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proceedings of the International Conference on Management of Data (ACM SIGMOD), pp. 1–12 (2000)
13. Deng, Z., Wang, Z., Jiang, J.: A new algorithm for fast mining frequent itemsets using N-lists. Sci. China Inf. Sci. **55**, 2008–2030 (2012)
14. Borah, A., Nath, B.: Rare pattern mining: challenges and future perspectives. Complex Intell. Syst. **5**, 1–23 (2019)
15. Ji, Y., Ying, H., Tran, J., Dews, P., Mansour, A., Massanari, R.M.: A method for mining infrequent causal associations and its application in finding adverse drug reaction signal Pairs. IEEE Trans. Knowl. Data Eng. **25**, 721–733 (2012)
16. Xu, T., Dong, X.: Mining frequent patterns with multiple minimum supports using basic Apriori. In: Natural Computation (ICNC), IEEE, pp. 957–961 (2013)
17. Kiran, R.U., Re, P.K.: An improved multiple minimum support based approach to mine rare association rules. In: Computational Intelligence and Data Mining (CIDM), IEEE, pp. 340–347 (2009)
18. Lee, Y.C., Hong, T.P., Lin, W.Y.: Mining association rules with multiple minimum supports using maximum constraints. Int. J. Approximate Reason. **40**, 44–54 (2005)
19. Darrab, S, David, B., Gunter, S.: Modern application and challenges for rare itemset mining. In: 8th International Conference on Knowledge Discovery (2019)
20. Fournier-Viger, P., Gomariz, A., Gueniche, T., Soltani A., Wu. C., Tseng V.S.: SPMF: a java open-source pattern mining library. J. Mach. Learn. Res. (JMLR), **15**, 3389–3393 (2014)
21. Darrab, S., Ergenç, B.: Frequent pattern mining under multiple support thresholds. In: The International Conference on Applied Computer Science (ACS), Wseas Transactions on Computer Research, pp. 1–10 (2016)
22. Darrab, S., Ergenç, B.: Vertical pattern mining algorithm for multiple support thresholds. In: International Conference on Knowledge Based and Intelligent Information and Engineering (KES), Procedia Computer Science, vol. 112, pp. 417–426 (2017)
23. Hu, Y.H., Chen, Y.L.: Mining association rules with multiple minimum supports: a new mining algorithm and a support tuning mechanism, Decision Support Systems, pp. 1–24 (2006)
24. Kiran, R.U., Reddy, P.K.: Novel techniques to reduce search space in multiple minimum supports-based frequent pattern mining algorithms. In: Proceedings of the International Conference on Extending Database Technology(EDBT), pp. 11–20 (2011)

Pareto Optimization in Oil Refinery

Dmitri Kostenko(✉), Dmitriy Arseniev, Vyacheslav Shkodyrev, and Vadim Onufriev

Peter the Great St. Petersburg Polytechnic University, Saint-Petersburg, Russia
Zaba-1@bk.ru

Abstract. This article describes the process of multicriteria optimization of a complex industrial control object using Pareto efficiency. The object is being decomposed and viewed as a hierarchy of embedded orgraphs. Performance indicators and controlling factors lists are created based on the orgraphs and technical specifications of an object, thus allowing to systematize sources of influence. Using statistical data archives to train, the neural network approximates key sensors data to identify the model of the controllable object and optimize it.

Keywords: Decomposition · Pareto efficiency · Multicriteria optimization · Identification · SPEA2 · Neural network

1 Introduction

Multicriteria optimization is a process of simultaneous optimization for two or more conflicting functions within one point [1]. The need to simultaneously optimize different key performance indicators (KPI) exists in both business and industrial production, such as oil refinement facilities.

Oil refinement is an advantageous direction for multicriteria optimization because of its complexity. Optimisation process is preceded by decomposition [2] of the refinement sequence, which is followed by model identification [3]. Pareto optimality principle [4] is used in conjunction with aforementioned models to build a front of optimal values.

Part of the refinement process takes place inside a refraction unit (RU). One of these units was taken as a prototype for our model. The model got the following KPIs assigned [5]: Quality (matching degree between output and established norms), Performance (output product volume), Efficiency (resource usage potency), Reliability (equipment failures per unit of time), Safety (emergencies per unit of time).

Rectification consists of a wide array of parameters, up to several hundreds of characteristics per one refraction unit. These include the splitting section, column head and column plates temperature and pressure. Inside and outside of the rectification column both sequential (multi-layered raw oil refinement, raw oil heating, raw oil pumping, etc.) and parallel (vapor condensation) processes take place. Rectification technology also includes transition products into the process, thus applying an additional, horizontal level of hierarchy between the operations. It is also essential to account for the time delay, added by the inertia of the system itself and enforced by the continuous operating mode.

Y. Tan et al. (Eds.): DMBD 2020, CCIS 1234, pp. 26–33, 2020.
https://doi.org/10.1007/978-981-15-7205-0_3

Consequently, processes from the highest levels of the hierarchy have unobvious connections with the lower-level processes thus making it impossible to use simple functions like y = f (g, u) to represent dependencies between them. Yet it is essential to influence top-level processes by changing parameters of the low-level processes and vice-versa.

In this work the aforementioned problem is resolved by decomposing a complex system (such as refraction unit) down to individual units and processes. The resulting structure is represented as a graph. The KPI set takes the top level of the hierarchy. Every KPI is divided into several summands of a lower hierarchy levels. The step is repeated until the summand can be unambiguously interpreted by the y = f(x) type of dependency. Dependencies are identified using a neural network trained on the RU statistical data archive. Going up by one hierarchy level changes dependency to y = g(f(x)). Ascending by the hierarchical tree allows to determinate a clean dependency between a KPI and an input parameter from the bottom of the hierarchy.

However, the top-level key performance indicators may directly contradict each other. For instance, raising Performance by forcing aggressive operating parameters will inevitably cause the growth in equipment failures and an overall reduction of Reliability. Which in turn damages Efficiency of the refraction unit or a refinery as a whole. The "Good – Fast – Cheap" triangle encourages us to use a multicriteria optimization algorithm to balance out conflicting key performance indicators.

The aim of this work is to perform a multicriteria optimization to find a Pareto optimal solution. This allows us to find a safe combination of controllable parameters able to keep the target indicators inside the given target intervals.

This analysis is based on statistical data archive taken from a working refinery. It was used to build the graphical representations of dependencies between performance and temperature, characterising the lowest hierarchy level. In order to optimize the top-level KPIs by changing the bottom-level controllable parameters, a strong correlation must be revealed. To grant it, a dependence model identification has been performed [6].

2 Data Identification

The refraction unit consists of an oil pre-heater with a heat-exchange unit, a fractionating column, a refrigerator and a boiler. The pre-heated oil is injected into the column feeding zone to be divided into vapour and solid phases. During the rectification process isopentane is extracted from the top part of the column as a fractionator overhead. Heavier fractions are taken from the plates in the middle of the rectification column. The heaviest part, the long residuum, gets extracted from the bottom part of the column [7]. A simplified scheme, showing distillation inputs and outputs targeted by this work, is present on Fig. 1.

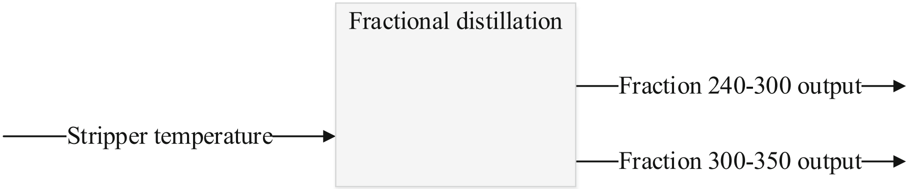


Fig. 1. Simplified scheme of the rectification process

The stripper temperature (U_4 in Table 1) has been chosen for optimization. Temperature variation was aimed at maximizing fractions 240–300 and 300–350 output volumes (G_2 and G_3 in Table 1).

Table 1. Hidden neural layer for fraction 240–300

Neyron	Input (IN)	Output (ON)	Input weight (IHN)	Output weight (OHN)
0	0,5329	0,4675	0,9646	0,5356
1	−0,4489	0,4218	−0,8125	−1,0897
2	0,6221	0,6083	1,1261	0,6041
3	−0,1149	0,5999	−0,2079	−0,7524
4	0,7458	0,5611	1,3499	0,8198
5	2,5314	0,7924	4,5817	3,1874
6	−1,0151	0,3031	−1,8373	−1,8294
7	1,5302	0,7177	2,7695	1,8181
8	−0,0096	0,4126	−0,0174	−0,3680
9	−0,3281	0,3505	−0,5939	−0,8191

To identify dependencies between the stripper temperature and fraction outputs (see Fig. 1), a neural network (NN) was utilized. It was trained on statistical data obtained during 24 h of the prototype column work.

The neural network consists of 1 input, 1 hidden and 1 output layers. Input and output layers both contain one neuron, while the hidden layer contains 10. The NN was trained by the backward propagation of errors method. Hidden and output layers use a sigmoid activation function (1):

$$F(x) = \frac{e^x}{e^x + 1} \tag{1}$$

The general formula for the NN is (2):

$$ON = f\left(\sum_{k=0}^{10} f\left(\sum_{k=0}^{10} IN * IHN\right) * OHN\right) \tag{2}$$

Here *f* stands for sigmoid activation function (1), while IN, ON, IHN and OHN are present in Tables 1 and 2.

Table 2. Hidden neural layer for fraction 300–350

Neyron	Input (IN)	Output (ON)	Input weight (IHN)	Output weight (OHN)
0	−0,4983	0,2103	−0,9457	0,7026
1	−3,5414	0,0843	−6,7410	−3,9722
2	−0,7764	0,3018	−1,4716	1,3219
3	0,7931	0,4216	1,5021	−1,4161
4	−0,7696	0,3039	−1,4587	1,3393
5	−0,1638	0,1935	−0,3109	0,2529
6	−0,1228	0,2843	−0,2331	0,1390
7	−0,2815	0,2045	−0,5342	0,4098
8	−0,5891	0,2605	−1,1171	0,9753
9	0,0778	0,2562	0,1475	−0,1713

Application of the aforementioned neural network to input data allowed to identify dependencies between the stripper temperature and output volumes of 240–300 and 300–350 fractions (see Fig. 2).

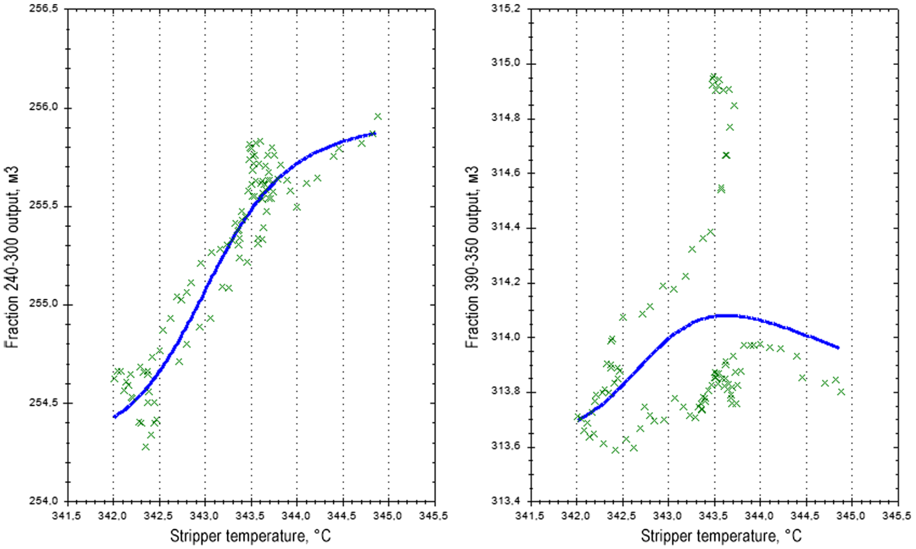


Fig. 2. Graphical representation of the identified models

To verify correlation between the models (lines on Fig. 3) and statistical data (crosses on Fig. 2) correlation coefficients $\rho_{x,y}$ have been calculated. The left graph $\rho_{x,y} = 0, 76312$ and the right graph $\rho_{x,y} = 0, 90781$. Coefficient formula (3) is present below, σ_x and σ_y are the mean values of the corresponding selections.

$$\rho_{x,y} = \frac{\frac{1}{n}\sum_{t=1}^{n} X_t Y_t - \left(\frac{1}{n}\sum_{t=1}^{n} X_t\right)\left(\frac{1}{n}\sum_{t=1}^{n} Y_t\right)}{\sigma_x * \sigma_y} \# \tag{3}$$

Correlation numbers could be improved by increasing the size of the training selection for the neural network. Amount of NN's hidden layers and/or hidden neurons are also subjects to change. But in order not to deviate from the main theme of the work, achieved correlations have been accepted.

Having mathematical models of interconnections between the basic parameters of the oil refinement process in place, it is now possible to compare them and define the optimal points according to the multicriteria optimization method.

3 Pareto Optimality

Pareto optimality is a state of allocation of resources from which it is impossible to reallocate so as to make any one individual or preference criterion better off without making at least one individual or preference criterion worse off [8].

Pareto front within the range of target functions is a combination of solutions, which do not dominate each other, but dominate every other solutions within the search space at the same time. It means that it is impossible to find a single solution able to excel every other solution at reaching every target. Mathematically such problem can be formulated as follows: one must find a vector $X^* = \left[x_1^*, x_2^*, \ldots, x_n^*\right]^{\mathrm{T}}$, that would optimize a vector of target functions $F(X) = \left[f_1(X), f_2(x), \ldots f_k(x)\right]^{\mathrm{T}}$ while having m inequality constraints $g_i(X) \leq 0, i = \overline{1, m}$, and p equality constraints $h_i(X) \leq 0, j = \overline{1, p}$.

Here $X^* \in R^n$ is a solution vector; $F(X) \in R^k$ is a vector of target functions every single one of which must be optimized [9].

Strength Pareto Evolutionary Algorithm 2 (SPEA2) [10] was used for Pareto optimization. Despite it's relatively old age, it is a well-tested algorithm, effective for select applications [11], including more representative spread of non-dominated solutions [12], and was chosen over others, including VEGA, FFGA [13] and NPGA [14].

SPEA2 algorithm can be summarized in 6 steps:

- Step 1, Initialization: Generate an initial population P_0 and create the empty archive (external set) $\overline{P_0} = \emptyset$. Set t = 0.
- Step 2, Fitness assignment: Calculate fitness values of individuals in P_t and $\overline{P_t}$.
- Step 3, Environmental selection: Copy all nondominated individuals in P_t and $\overline{P_t}$ to P_{t+1}. If size of P_{t+1} exceeds $\overline{N}$, then reduce P_{t+1} by means of the truncation operator, otherwise if size of P_{t+1} is less than $\overline{N}$, then fill P_{t+1} with dominated individuals in P_t and $\overline{P_t}$.
- Step 4, Termination: If t ≥ T or another stopping criterion is satisfied then set A to the set of decision vectors represented by the nondominated individuals in P_{t+1}. Stop.

- Step 5, Mating selection: Perform binary tournament selection with replacement on P_{t+1} in order to fill the mating pool.
- Step 6, Variation: Apply recombination and mutation operators to the mating pool and set P_{t+1} to the resulting population. Increment generation counter (t = t + 1) and go to Step 2.

SPEA2 algorithm has been utilized to find a front of temperatures, maximizing the output of fractions 240–300 and 300–350. Previously identified models have been used as "experimental data" on the graph seen on Fig. 3.

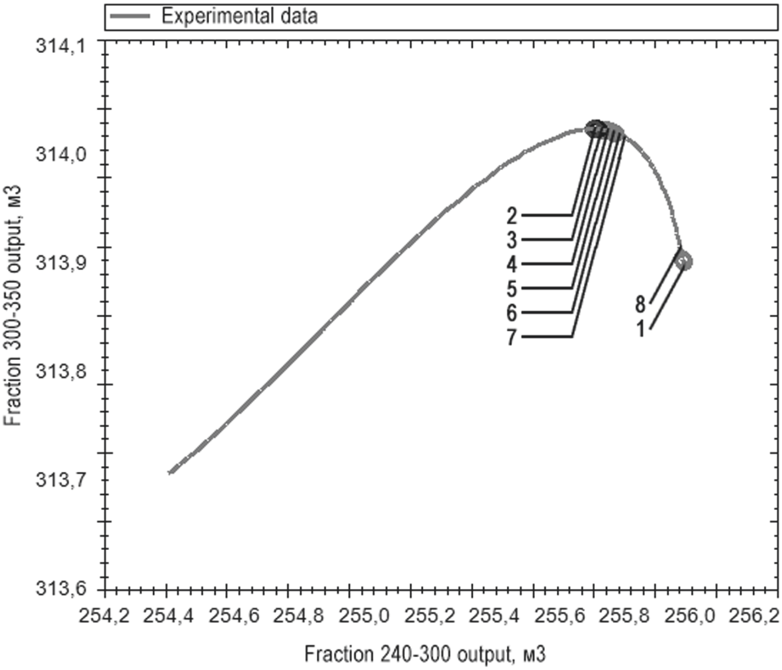


Fig. 3. Pareto front

As a result, we got a Pareto front, consisting of eight points (see Table 3). The points are sorted in descending order of preference, thus making the first point the most optimal.

Table 3. Complete stats of the Pareto front

Point number	Temperature (°C)	Fraction 240–300 output volume (M^3)	Fraction 300–350 output volume (M^3)
1	344,823852539063	255,869043673089	313,914703058703
2	343,325988769531	255,623350989602	314,04678075533

(*continued*)

Table 3. (*continued*)

Point number	Temperature (°C)	Fraction 240–300 output volume (M^3)	Fraction 300–350 output volume (M^3)
3	343,371765136719	255,636549610692	314,046650398148
4	344,712951660156	255,866950335304	313,918779855307
5	343,374877929688	255,661217222795	314,045467958112
6	343,35595703125	255,672724869121	314,044445521759
7	343,364624023438	255,649165327318	314,046207859264
8	343,407836914063	255,683708140522	314,043155299305

4 Conclusion

In the course of the work a refraction unit has been decomposed, and it's input, controllable and target parameters were extracted and listed. Based on statistical data analysis the neural network was trained and then used to identify dependency models between the RU parameters. The models allowed us to define eight points of Pareto front.

We are planning on extend the sphere of practical application of this method. First of all, we'll have to decompose complete sets of dependencies from the basic controllable parameters up to top level KPIs. This would allow us to see key performance indicators of separate units and the whole refinery changing in real time.

Further software development enables us to revert the process and guess controllable parameters, able to sustain a pre-defined set of high-level KPIs. Such instrument will not only allow to optimize complex processes, but will also ensure much more effective control over them.

References

1. Steuer, R.: Multiple Criteria Optimization: Theory, Computations, and Application. Wiley, New-York (1986)
2. Yaochu, J.: Pareto-optimality is everywhere: From engineering design, machine learning, to biological systems. In: 2008 3rd International Workshop on Genetic and Evolving Systems, p. 1. IEEE (2008)
3. Kostenko, D., Kudryashov, N., Maystrishin, M., Onufriev, V., Potekhin, V., Vasiliev, A.: Digital twin applications: diagnostics, optimisation and prediction. In: Proceedings of the 29th DAAAM International Symposium, pp. 574–581. DAAAM International, Vienna (2018)
4. Evans, G.W., Stuckman, B., Mollaghasemi, M.: Multicriteria optimization of simulation models. In: 1991 Winter Simulation Conference Proceedings, pp. 894–900. Phoenix, Arizona (1991)
5. Key performance indicators in the oil & gas industry. https://www.performancemagazine.org/key-performance-indicators-oil-bp/. Accessed 23 Jan 2019
6. Schleicha, B., Anwer, N., Mathieu, L., Wartzack, S.: Shaping the digital twin for design and production engineering. CIRP Ann. – Manuf. Technol. **66**(1), 141–144 (2017)

7. Alfke, G., Irion, W.W., Neuwirth, O.S.: Oil refining. In: Ullmann's Encyclopedia of Industrial Chemistry (2007)
8. Keeney, R., Raiffa, H.: Decisions with Multiple Objectives: Preferences and Value Trade-Offs. Cambridge University Press, Cambridge (1993)
9. Coello Coello, C.A., Christiansen, A.D.: Multi-objective optimization of trusses using genetic algorithms. Comput. Struct. **75**(6), 647–660 (2000)
10. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In: TIK report № 103. Switzerland, Zurich (2001)
11. Tang, Y., Reed, P., Wagener, T.: How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration? Hydrol. Earth Syst. Sci. Discuss. Eur. Geosci. Union **10**(2), 289–307 (2006)
12. Chołodowicz, E., Orlowski, P.: Comparison of SPEA2 and NSGA-II applied to automatic inventory control system using hypervolume indicator. Stud. Inform. Control **26**(1), 67–74 (2017)
13. Fonseca, C.M., Fleming, P.J.: Genetic algorithm for multiobjective optimization, formulation, discussion and generalization. In: Genetic Algorithms: Proceeding of the Fifth International Conference, California, pp. 416–423 (1993)
14. Horn, J.N., Nafpliotis, A.L., Goldberg, D.E.: A niched Pareto genetic algorithm for multiobjective optimization. In: Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, pp. 82–87. IEEE Service Center, Piscataway (1994)

Hypothesis Verification for Designing Flyback Booster by Analysis of Variance Visualized on Triangular Matrix Representations

Kazuhisa Chiba[1](✉), Taiki Hatta[1], and Masahiro Kanazaki[2]

[1] The University of Electro-Communications,
1-5-1, Chofugaoka, Chofu, Tokyo 182-8585, Japan
kazchiba@uec.ac.jp
[2] Tokyo Metropolitan University,
6-6, Asahigaoka, Hino, Tokyo 191-0065, Japan
kana@tmu.ac.jp
http://www.di.mi.uec.ac.jp/chiba/index_e.html

Abstract. This study performed data mining for nondominated-solution datasets of flyback-booster geometry for next-generation space transportation procured by evolutionary computation. We prepared two datasets of nondominated solutions due to two problem definitions, which differ merely in the definition of some design variables based on a design hypothesis gained from evolutionary-computation results. This study aims at verifying the hypothesis by applying mining to these two datasets to elucidate the contrast in the influence of the design variables. We used functional analysis of variance for data mining; scrutinized the effects of single and two-combined design variables. Furthermore, intuitive visualization by triangular matrix representations could distinguish the discrepancy between the obtained results. The consequence has verified the significance of the hypothesis; it revealed that the discontinuous surface naturally evaded in the hypersonic range because of surface temperature upsurge is capable of enhancing the lift-to-drag ratio in the low-speed range; the hypothesis grew into a new design problem.

Keywords: Design informatics · Space transportation · Information visualization · Real-world design problems

1 Introduction

High-frequency and price reduction of space transportation are urgent priorities by the recent gains in the demands of small satellites. Under this circumstance, reusable launch systems (RLSs) are one of the realizability in several aspects to attain the purposes as a successor of the Space Shuttle. Hence, studies have

Supported by JHPCN, JSPS KAKENHI, and IFS Tohoku University.

Y. Tan et al. (Eds.): DMBD 2020, CCIS 1234, pp. 34–44, 2020.
https://doi.org/10.1007/978-981-15-7205-0_4

lately activated again for developing reusable space transportation, mainly in China [16], Russia [11], the E.U. [14], and the U.S [7]. Meanwhile, Japan is currently conducting a collaborative research project on two-stage RLS among some universities. Launch experiments of a prototype model are underway to establish a control law for launching an authentic vehicle [6]. The project is designing trajectories and body geometries at the same time; namely, we are studying the conceptual design for an RLS.

The 1st evolutionary optimization for 3D geometry design accidentally created a discontinuous ditch on the back of flyback boosters [15]. The hypersonic region covered in the trajectory generally eludes the discontinuity of body surface that causes the temperature to rise due to adiabatic compression. Therefore, the geometry representation unit in the optimal design system solved the inverse problem and modified the discontinuous ditch intentionally to make them continuous. When we evolved the 2nd optimization in the equal generation number using the identical initial population as the 1st optimization, the lift-to-drag ratio (L/D) in the transonic/supersonic range declined wholly [8]. This result implies the hypothesis for the body design that achieves moderate accomplishments while improving the aerodynamic performance in the low-speed region by the deliberate discontinuity of the body surface promoting the vortex lift.

This study aims to verify the design hypothesis by performing data mining on nondominated-solution sets of flyback-booster geometry procured by evolutionary computations (ECs) and elucidating the contribution of design variables to the L/D, namely the study is an application of data mining to real-world design. The mining target is two sets of nondominated solutions gained for two problems that vary simply in the definition of some design variables based on the design hypothesis. We scrutinize distinction in the influence of design variables on the L/D. Since the difference in the optimization results revealed an apparent decrease in the L/D, it infers that the character of surface ditches would emerge not to be small; the data mining will utilize a functional analysis of variance to examine each design variable and combinations between two design variables thoroughly. Furthermore, visualization by triangular matrix representations (TMR) [1] will facilitate intuitive perceptions of the procured distinctions.

We structure this article as follows. Section 2 explains the problem settings and outlines the ways of data mining concisely. Section 3 reviews the design information caused by the optimization. In Sect. 4, we assess the gained results by data mining. Section 5 concludes this article.

2 Problem Settings

At present, we are promoting research on space transportation systems at several domestic universities, including Kyushu Institute of Technology, which designs and develops fully reusable space transportation WIRES [6]. We tried to design space transportation, commencing from the flight path optimized with WIRES. Initially, although the optimum trajectory also alters according to the change of the geometry, the hurdle for generating the aerodynamic performance matrix necessary for trajectory optimization is high, so it is a future work.

Table 1. Definition of design variables.

Section number	Design-variable S/N number	Symbol (refer to Fig. 2)
②	1, 3, 5, 7, 9	V_{y1}, V_{y2}, V_{y3}, V_{y4}, V_{y5}
	2, 4, 6, 8, 10	V_{z1}, V_{z2}, V_{z3}, V_{z4}, V_{z5}
③	11, 13, 15, 17, 19	V_{y1}, V_{y2}, V_{y3}, V_{y4}, V_{y5}
	12	V_{z1}
	14, 16, 18	V_{z2}, V_{z3}, V_{z4}
	20	V_{z5}
④	21, 23, 25, 27, 29	V_{y1}, V_{y2}, V_{y3}, V_{y4}, V_{y5}
	22	V_{z1}
	24, 26, 28,	V_{z2}, V_{z3}, V_{z4}
	30	V_{z5}
⑤	31, 33, 35, 37, 39	V_{y1}, V_{y2}, V_{y3}, V_{y4}, V_{y5}
	32,	V_{z1}
	34, 36, 38,	V_{z2}, V_{z3}, V_{z4}
	40	V_{z5}

The trajectory used in this study assumes an injection of 10 t payload into the orbit from the Tanegashima Space Center and a circular orbit at an altitude of 350 km. Based on this, they were defining the aerodynamic performance optimization in three points of transonic, supersonic, and hypersonic speeds. At the defined transonic/supersonic design point, the booster is the point where we have to obtain a range for fly back to the launch field. At the hypersonic design point where a booster and an orbiter separate, at this point, heave the altitude and take a sequence to secure the range margin.

2.1 Optimization Problem

This problem defines six objective functions: three for aerodynamics (maximizing the L/D at $M = 0.65$, 2.3, and 6.8), one for the structures (minimizing empty weight), and two for aerodynamic heating (minimizing body surface temperature and body surface area where thermal protection systems attaches).

As shown in Fig. 1, we generate six cross-sectional shapes for the x axis direction. We utilize section numbers ① and ⑥ only to satisfy later-described constraint conditions, and the cross-sectional shape change in optimization is four cross-sections with ② to ⑤. Since one cross-section uses ten design variables as shown in Fig. 2, the total of design variables is 40. Table 1 explains the definition of each design variable.

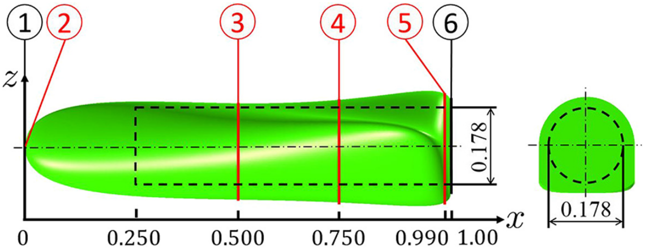
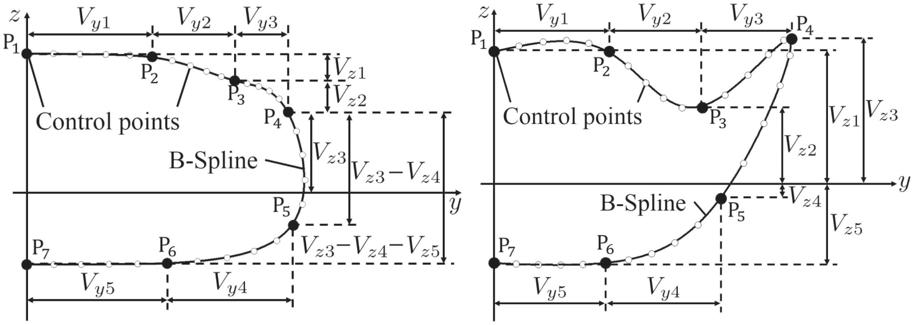


Fig. 1. Left: cross-section positions, right: tail surface. The dotted line describes a fuel tank to be a fixed size.



(a) cross-section②

(b) cross-sections③to⑤

Fig. 2. Relations among design variables V_n and control points P_m at each cross-section. White points from P_i to P_{i+1} are the subordinate control points of P_m.

We define constraints on the geometry; no constraint provides for the objective functions. When making individuals satisfying the geometrical constraints for the population size, the optimization system ceases creating ten individuals for one generation.

2.2 Optimizer

Since one of the information desired by the multiobjective optimization is the executable structure of the objective-function space, we employed an EC that performs a global search for the optimization method; we made a fully automated design system [2]. This study adopted the strength Pareto evolutionary algorithm 2 (SPEA2) [17] in many ECs due to conventionality. We respectively chose simulated binary crossover [4] and polynomial mutation [5] in the standard.

2.3 Data-Mining Technique

A functional analysis of variance (ANOVA) quantifies the contribution rates to the variance of a model [9]. ANOVA decomposes the total variance of the model into that of each design variable and their interactions to estimate their impact quantitatively by integrating values out of the model $\widehat{y}$. The Kriging model [10] for an objective value $y(\boldsymbol{x})$ of design variables $\boldsymbol{x}$ ($\dim \boldsymbol{x} = N$)

$$y(\boldsymbol{x}) = \beta + Z(\boldsymbol{x}) \tag{1}$$

predicts the model $\widehat{y}$. β and $Z(\boldsymbol{x})$ are a constant term of the model and a Gaussian random process with zero mean and variance of σ^2, respectively. In this case, the single effect of x_i is

$$\begin{aligned}\widehat{\mu}(x_i) = &- \underbrace{\int_{D^N} \widehat{y}(\boldsymbol{x}) \mathrm{d}x_1 \mathrm{d}x_2 \cdots \mathrm{d}x_N}_{\widehat{\mu}(\boldsymbol{x})} \\ &+ \int_{D^{N-1}} \widehat{y}(\boldsymbol{x}) \mathrm{d}x_1 \mathrm{d}x_2 \cdots \mathrm{d}x_{i-1} \mathrm{d}x_{i+1} \cdots \mathrm{d}x_N\end{aligned} \tag{2}$$

and likewise the combined impact of x_i and x_j $(i < j)$ is

$$\begin{aligned}\widehat{\mu}(x_{i,\,j}) = &- \widehat{\mu}(\boldsymbol{x}) - \widehat{\mu}(x_i) - \widehat{\mu}(x_j) \\ &+ \int_{D^{N-2}} \widehat{y}(\boldsymbol{x}) \mathrm{d}x_1 \mathrm{d}x_2 \cdots \mathrm{d}x_{i-1} \mathrm{d}x_{i+1} \cdots \mathrm{d}x_{j-1} \mathrm{d}x_{j+1} \cdots \mathrm{d}x_N .\end{aligned} \tag{3}$$

The variance $\widehat{\varepsilon}(x_i)$ caused by x_i is

$$\widehat{\varepsilon}(x_i) = \int \left(\widehat{\mu}(x_i)\right)^2 \mathrm{d}x_i, \tag{4}$$

so

$$p(x_i) = \frac{\widehat{\varepsilon}(x_i)}{\widehat{\varepsilon}(\boldsymbol{x})} \tag{5}$$

distributes the proportion on the contribution of x_i.

Various information visualization techniques express database search results as sturdy devices for knowledge discovery [12]. The filtering techniques, including TMR [1], have proven to be serviceable for tasks where visual representations can find relationships, clusters, outliers, gaps, and other patterns [13].

3 Review of the Optimization Results

This section concisely reviews the knowledge that the optimizations have already indicated, which is crucial for interpreting the data-mining results.

3.1 Knowledge from the Optimization Results

The ditch brings a positive effect in the lift; it causes a negative influence on the drag. As a result of the optimization, we anticipate that the ditch will have a positive impact on the low-speed range L/D. Therefore, this study focuses on improving the low-speed range L/D and does not discuss the result at $M = 6.8$.

Difference of Design Strategies. Optimal geometries of each objective function for the L/D explicates the following diversity in design strategies.

- L/D at $M = 0.65$ grows by raising the lift.
- L/D at $M = 6.8$ rises by reducing the drag.
  The optimization makes the body slender to reduce the drag in the hypersonic range. Since slim bodies concomitantly enable to decrease the surface temperature, they simultaneously improve three of the six objective functions; it must stipulate the evolutionary trend.
- Strategy for increasing the L/D at $M = 2.3$ is related to that for the L/D at $M = 6.8$, but with scope for a rising lift.

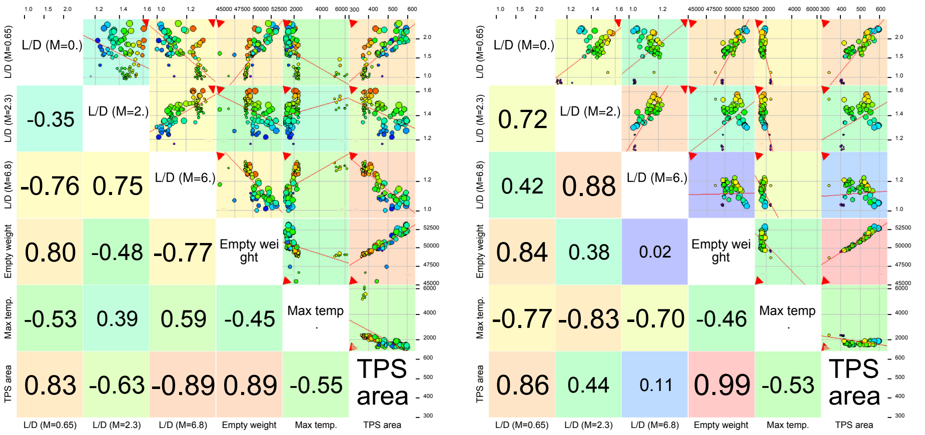
Influence of Ditch

- Positive effect in the lift: ditches facilitate vortex form; they grow vortex lift.
- Negative influence on the drag: the induced drag rises due to the vortex lift; the friction drag first increases with the expansion of the wetted area.

3.2 Design Information from SPM

The 1st and 2nd optimization results shown in Fig. 3 present the effect of the presence or absence of a ditch on the transonic/supersonic L/D as follows.

- To eliminate ditch deteriorates the L/D at $M = 0.65$, approximately 5.4%.
- To eliminate ditch deteriorates the L/D at $M = 2.3$, approximately 1.9%.
- The L/D at $M = 6.8$ also declines, but it links to the L/D at $M = 2.3$.



(a) 1st result considering ditch.

(b) 2nd result without considering ditch.

Fig. 3. Scatter plot matrix of nondominated solutions for 30 generations by SPEA2 in the objective-function space. Red lines/triangles signify regression lines/optimum directions. Numbers on lower triangular matrices specify correlation coefficients. (Color figure online)

Originally, $M = 0.65$ is more reliant on the lift than $M = 2.3$ for improving the L/D, so we infer that to add ditch produces a more positive effect on the lift than a negative impact on the drag.

4 Results by Data Mining

The purpose of data mining is to verify whether the accidentally created ditches on the back of boosters contribute to improving L/D. So, we have to prove the following two perspectives: (1) the dataset of 1st optimization result that considers ditches affects design variables involved in ditch generation, (2) the dataset of 2nd optimization result that excludes ditches does not impact design variables required for ditch representation. Note that the results of ANOVA gained for each objective function quantify the contribution of each design variable, assuming that the total influence on an objective function is 100%. We cannot quantitatively discuss the relative effects among objective functions.

4.1 Result for Dataset with Ditch

***L/D* at *M* = 0.65**. TMR shown in Fig. 4(a) indicates that the design variables with substantial influence are $dv34$, $dv37$, and $dv40$ as well as with weak effect are $dv12$, $dv20$, $dv22$, and $dv27$. The beneficial design variables register that the design strategies for improving L/D at $M = 0.65$ are (a) spreading the body in the spanwise direction and making the bottom as flat as possible, and (b) providing ditches. To prolong the wetted area, that is, to stretch the body in the spanwise direction is the most valuable for raising the lift, which effects on improving L/D at $M = 0.65$. Therefore, the influence of the design variables ($dv20$, $dv27$, $dv37$, $dv40$) for expanding the fuselage in the spanwise direction and flattening the bottom surface is relatively more massive than the design variables ($dv12$, $dv22$, $dv34$) for providing ditches. However, we can affirm that the provision of the ditch is undoubtedly useful in improving the L/D.

1. $dv34$ ($V_{z2}^{\textcircled{5}}$[1]): to raise the difference from V_{z1} boosts the curvature of the B-Spline curve from P_1 to P_3, resulting in a deeper ditch; the ditch affects the low-speed L/D. Note that V_{z1} tends to be large to serve the constraint of securing the tank volume, so V_{z2} inclines to be small.
2. $dv37$ ($V_{y4}^{\textcircled{5}}$): decreasing this value widens the flat bottom region in the span direction; growing the pressure on the bottom of the fuselage gains the lift.
3. $dv40$ ($V_{z5}^{\textcircled{5}}$): by reducing the bottom thickness and flattening fuselage in the z-direction, the base matures a flat geometry. The identical design concept as $dv37$ contributes to lift repair.

[1] For example, $dv34$ symbolizes the design variable V_{z2} of the fifth cross-section displayed in Table 1 and Fig. 2, so this paper expresses it as $V_{z2}^{\textcircled{5}}$.

***L*/*D* at *M* = 2.3.** TMR shown in Fig. 4(b) indicates that the design variables with substantial influence are $dv14$, $dv17$, $dv22$, and $dv40$. Since drag alleviation helps improve L/D at $M = 2.3$, unlike optimizing the L/D at $M = 0.65$, no geometry widens horizontally and tends to be elongated. Hence, the impact of the design variables ($dv17$, $dv40$) that expand the body horizontally and the design variables ($dv14$, $dv22$) that supplement ditches to the lift is relatively robust in the latter.

4.2 Result for Dataset Without Ditch

***L*/*D* at *M* = 0.65.** Since all design variables indicated by TMR shown in Fig. 5(a) are irrelevant to ditches, ditches do not affect on the L/D. The abilities of design variables that have direct and indirect effects are abstracted below.

- design variables with direct effects
  1. $dv10$: $V_{z5}^{\textcircled{2}}$ has the effect of gaining the lift by making the bottom of the body nose as flat as possible.
  2. $dv35$: $V_{y3}^{\textcircled{5}}$ extends the area near the body tail in the spanwise direction and creates a stabilizer to enhance the lift.
- design variables with indirect effects
  1. $dv11$, $dv23$: $V_{y1}^{\textcircled{3}}$ and $V_{y2}^{\textcircled{4}}$ widen the body sideways and gain the lift.
  2. $dv40$: $V_{z5}^{\textcircled{5}}$ raises the lift by flattening the bottom near the body tail.

***L*/*D* at *M* = 2.3.** All design variables indicated by TMR shown in Fig. 5(b) are independent of the ditch, so the impact of the ditch faded away in the L/D at $M = 2.3$ as well as in the L/D at $M = 0.65$. The capabilities of design variables that affect the L/D at $M = 2.3$, as shown by TMR, are as follows.

- design variables with direct influences
  1. $dv10$: $V_{z5}^{\textcircled{2}}$ is a variable that ultimately transforms the L/D because it is useful in raising the lift by flattening the bottom of the body's nose and in diminishing the drag by building the curvature.
- design variables with indirect effects
  1. $dv8$, $dv20$, $dv40$: $V_{z4}^{\textcircled{2}}$, $V_{z5}^{\textcircled{3}}$, and $V_{z5}^{\textcircled{5}}$ have effects comparable to $dv10$.
  2. $dv22$ ($V_{z1}^{\textcircled{4}}$): the direct effect is weak, and the result is more robust when combined with $dv10$, which implies that the cross-sectional shape transformation in the flow direction affects the L/D. Generally, in the supersonic range, the cross-sectional area distribution of the body in the flow direction desires alteration monotonically from the sonic boom theory [3]. The combination of $dv10$ and $dv22$ must deviate from a monotonic increase, leading to a rise in the wave drag, resulting in the L/D deterioration. In particular, since $dv22$ represents the cross-section $\textcircled{4}$, its cross-section is likely to disturb area distribution because there commences forming a stabilizer. In any case, $dv22$ does not affect the lift.

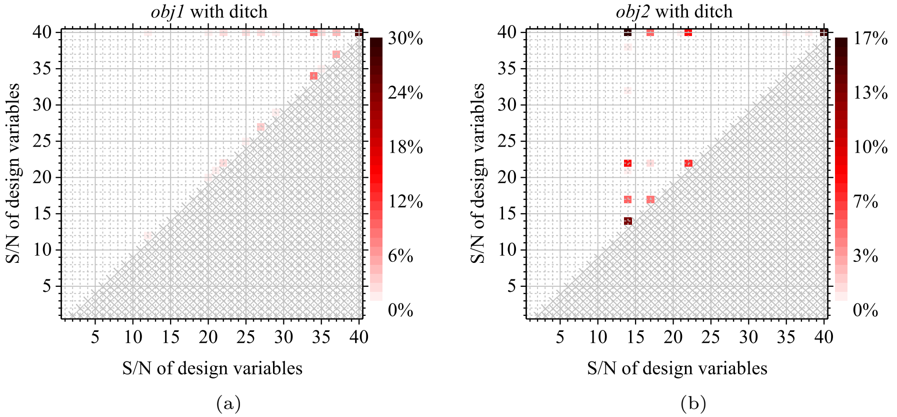


Fig. 4. TMR of ANOVA results for the dataset of the 1st optimization, which accidentally deals with ditches. (a) L/D at $M = 0.65$ and (b) L/D at $M = 2.3$.

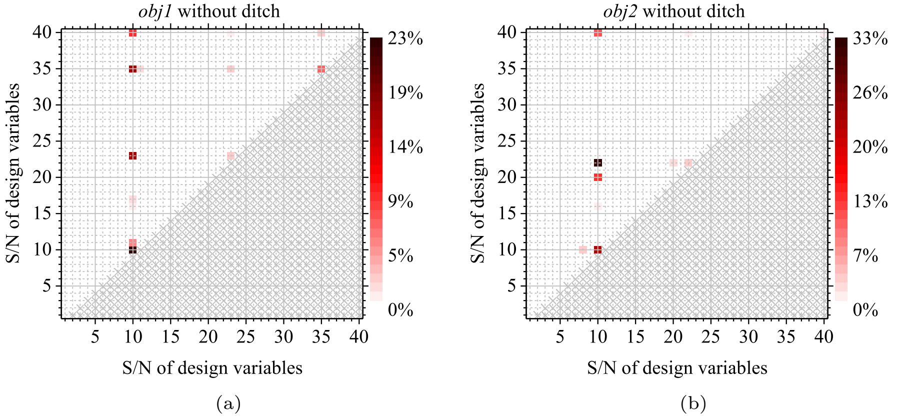


Fig. 5. TMR of ANOVA results for the dataset of the 2nd optimization, which solves the inverse problem to eliminate ditches. (a) L/D at $M = 0.65$ and (b) L/D at $M = 2.3$.

5 Conclusion

This research performed data mining by functional analysis of variance on multidisciplinary many-objective optimization results for a real-world problem and intuitively visualized the mining results using triangular matrix representations. We aimed at verifying a design hypothesis for a flyback booster in space transportation. Ditches avert surface temperature surge caused by adiabatic compression in the hypersonic range. However, they on the body back facilitated the lift rise in the transonic and supersonic ranges. We proved that ditches contributed to the increment in the low-speed lift-to-drag ratio; the study has affirmed that the hypothesis was correct. We have consequently gained a new design problem that examines how to add ditches to the body surface.

References

1. Birkenmeier, G.F., Heatherly, H.E., Kim, J.Y., Park, J.K.: Triangular matrix representations. J. Algebra **230**(2), 558–595 (2000). https://doi.org/10.1006/jabr.2000.8328
2. Chiba, K., Sumimoto, T., Sawahara, M.: Completely automated system for evolutionary design optimization with unstructured computational fluid dynamics. In: Proceedings of International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence. ACM (2019). https://doi.org/10.1145/3325773.3325778
3. Darden, C.M.: Sonic boom theory - its status in prediction and minimization. J. Aircr. **14**, 569–576 (1977)
4. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex Syst. **9**(2), 115–148 (1995)
5. Deb, K., Agrawal, R.B.: A combined genetic adaptive search (GeneAS) for engineering design. Comput. Sci. Inform. **26**, 30–45 (1996)
6. Fujikawa, T., et al.: Research and development of winged reusable rocket: current status of experimental vehicles and future plans. In: Proceedings on Asia-Pacific International Symposium on Aerospace Technology. JSASS, Soul, Republic of Korea (2017)
7. Harris, M.: The heavy lift: blue origin's next rocket engine could power our return to the moon. IEEE Spectr. **56**, 26–30 (2019). https://doi.org/10.1109/MSPEC.2019.8747308
8. Hatta, T., Sawahara, M., Chiba, K.: Many-objective multidisciplinary evolutionary design for hybrid-wing-body-type flyback booster on an entirely automated system. In: Proceedings on International Conference on Evolutionary and Deterministic Methods for Design, Optimization, and Control with Applications to Industrial and Societal Problems 2019. ECCOMAS (2019)
9. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. J. Global Optim. **13**(4), 455–492 (1998)
10. Keane, A.J.: Wing optimization using design of experiment, response surface, and data fusion methods. J. Aircr. **40**(4), 741–750 (2003)
11. Kolychev, A.V., Kernozhitskii, V.A., Chernyshov, M.V.: Thermionic methods of cooling for thermostressed elements of advanced reusable launch vehicles. Russ. Aeronaut. **62**(4), 669–674 (2019). https://doi.org/10.3103/S1068799819040184
12. Sacha, D., Stoffel, A., Stoffel, F., Kwon, B.C., Ellis, G., Keim, D.A.: Knowledge generation model for visual analytics. IEEE Trans. Vis. Comput. Graph. **20**(12), 1604–1613 (2014)
13. Shneiderman, B.: Extreme visualization: squeezing a billion records into a million pixels. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 3–12 (2008). https://doi.org/10.1145/1376616.1376618
14. Simplício, P., Marcos, A., Bennani, S.: Reusable launchers: development of a coupled flight mechanics, guidance, and control benchmark. J. Spacecr. Rockets **56**, 74–89 (2019). https://doi.org/10.2514/1.A34429
15. Sumimoto, T., Chiba, K., Kanazaki, M., Fujikawa, T., Yonemoto, K., Hamada, N.: Evolutionary multidisciplinary design optimization of blended-wing-body-type flyback booster. In: AIAA Paper 2019–0703 on the 57th AIAA Aerospace Science Meeting. AIAA (2019)

16. Zhou, H., Wang, X., Cui, N.: Glide guidance for reusable launch vehicles using analytical dynamics. Aerosp. Sci. Technol. **98**, 1–2 (2020). https://doi.org/10.1016/j.ast.2019.105678
17. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength Pareto evolutionary algorithm. TIK-Report No. 103, Computer Engineering and Communication Networks Lab., ETH Zurich (2001)

Application of Decision Tree Algorithm Based on Clustering and Entropy Method Level Division for Regional Economic Index Selection

Yi Zhang(✉) and Gang Yang

Department of Information and Software Engineering,
Chengdu Neusoft University,
Dujianyan, Chengdu 611844, Sichuan, China
zy1044911641@gmail.com

Abstract. The economy of a region is affected by many factors. The purpose of this study is to use the entropy method clustering and decision tree model fusion to find the main factors affecting the regional economy with the support of big data and empirical evidence. First extract some important indicators that affect the regional economy, and use the entropy method to find the relative weights and scores of these indicators. Then use K-means to divide these indicators into several intervals. Based on the entropy fusion model, obtain the ranking of each category of indicators, use these rankings as the objective value of the decision tree, and finally establish an economic indicator screening model. Participate in optimization and build a decision tree model that affects regional economic indicators. Through the visualization of the tree and the analysis of feature importance, you can intuitively see the main indicators that affect the regional economy, thereby achieving the research goals.

Keywords: Clustering · Entropy evaluation method · Decision tree · Regional enconomy · Random forest

1 Introduction

The development of a regional economy is related to many factors, which will positively or negatively affect the development of the regional economy. The entropy method is an objective weighting method. The entropy value is used to judge the degree of discreteness of an indicator The greater the degree, the greater the impact (weight) of the indicator on the comprehensive evaluation, and the smaller its entropy value. In the improved entropy method and its application in the evaluation of economic benefits [4], the power factor method mentioned in the article solves some extreme data and indicators but essentially uses the entropy method to evaluate regional economic benefits. Objective It is relatively strong, and lacks certain persuasion for larger data samples and more

Y. Tan et al. (Eds.): DMBD 2020, CCIS 1234, pp. 45–56, 2020.
https://doi.org/10.1007/978-981-15-7205-0_5

economic indicators. At the same time, the basic use of the CART decision tree was proposed in [5], but the application of index screening in the evaluation of the regional economy was still lacking. In this paper, we propose the concept of model fusion that combines CART decision tree and clustering machine learning algorithms for regional economic level evaluation and division based on the use of entropy method, which can more accurately extract the indicators that affect economic effects. As described below.

The entropy method objectively calculates the corresponding weight score based on the degree of discreteness of each feature data, and the distribution of the data is basically discrete, and the data is huge. Here we choose the K-means clustering algorithm based on the division to classify, on large data sets The calculation efficiency is also very high, and then the weight score is used to calculate the average score of each category, and the classification is divided. Because most of the data types are continuous, the CART decision tree is selected as the classification model. However, on the huge data, although the corresponding parameter adjustment and optimization operations are performed, it is inevitable that there will be overfitting and generalization capabilities. Worse. The optimization problem uses the random forest in ensemble learning for optimization, which can not only effectively run on big data but also solve high-dimensional data problems without reducing the dimension, and the estimated model is an unbiased model. To this end, we use several types of algorithm model fusion and use k-fold cross-validation to evaluate and tune the model. Our experiments on the economic data set opened by the Singapore government show that the accuracy rate of the final optimized model fusion is as high as 94%, Analyzed the main indicators affecting regional economic development, and put forward some suggestions to help regional economic sustainable development.

The rest of this paper is organized as follows: Sect. 2 discusses the main research methods, In Sect. 3 discussed the establishment of models, how to carry out model fusion, Corresponding experiments are carried out in Sect. 4, and the cross-validation results are given at the end. Finally, the work is concluded in Sect. 5.

2 Research Methods

2.1 Introduction to K-Means Clustering Algorithm

The k-means clustering algorithm is an iterative solution based partitioning cluster analysis algorithm. It uses distance as a standard for measuring the similarity between data objects. Euclidean distance is usually used to calculate the distance between data objects. The formula for calculating the Euclidean distance is given below 1:

$$\operatorname{dist}(x_i, x_j) = \sqrt{\sum_{d=1}^{D} (x_{i,d} - x_{j,d})^2} \tag{1}$$

Complete k-means algorithm flow, such as Algorithm 1

Algorithm 1: K-means clustering algorithm

```
Seeding: $k$ initial centers $\mathcal{C} = \{c_1, c_2, \ldots, c_k\}$
repeat
    foreach $i \in \{1, \ldots, k\}$ do
        $C_i \leftarrow \{x \in \mathcal{X} \mid \|x - c_i\| < \|x - c_j\| \, \forall j \neq i\}$ ;    / * $C_i$ is assigned
        the set of all points in $\mathcal{X}$ having $c_i$ as their closest center */
    end
    foreach $i \in \{1, \ldots, k\}$ do
        $c_i \leftarrow \frac{1}{|C_i|} \sum_{x \in C_i} x$;    /* modify $c_i$ to be the center of mass of $C_i$
        */
    end
until no more change of $C$;
return $\mathcal{C}$;
```

2.2 An Introduction to the Entropy Method

The entropy method refers to a mathematical method used to judge the degree of dispersion of a certain index. The greater the degree of dispersion, the greater the impact of this indicator on comprehensive evaluation. You can use the entropy value to judge the degree of dispersion of a certain index because, the information entropy can be used to calculate the weight of each indicator according to the degree of variation of each indicator, which provides a basis for comprehensive evaluation of multiple indicators.

Information entropy refers to the concept of entropy in thermodynamics and describes the average information of the source, as shown in formula 2

$$H(x) = E\left[\log \frac{1}{p\left(a_i\right)}\right] = -\sum_{i=1}^{q} P\left(a_i\right) \log P\left(a_i\right) \tag{2}$$

2.3 Overview of Decision Tree Generation Algorithms

The entire decision tree is based on the tree structure to make decisions, which can be divided into three progressive processes: optimal feature selection, decision tree generation, and pruning. The internal node corresponds to a test on an attribute, each branch corresponds to a possible result of the test, that is, a certain value of the attribute, and each leaf node corresponds to a prediction result. The main information gain calculation method is as follows:

$$IG(S|T) = \text{Entropy}(S) - \sum_{value(T)} \frac{\left(S_v\right)}{S} \text{Entropy}\left(S_v\right) \tag{3}$$

The term before the minus sign in the formula is the entropy of the training set classification, S represents the sample set, T is the set of all feature values, and Sv is the feature equal to v in the feature; the second half of the minus sign is the entropy for classification with v. The subtraction of the two entropies has the following meaning: using this feature classification, it can reduce how much uncertainty and how much information is carried.

2.4 Random Forest Algorithm Based on CART Tree

CART can be used for both regression analysis and classification analysis, and some integrated algorithms based on CART have been extended. To solve the problem of large data size and data volume in the context of big data, this study chose CART as the Basic random forest algorithm.

The CART decision [3] tree has the advantages of being easy to understand and having certain non-linear classification capabilities, but a single decision tree has some disadvantages.

The above-mentioned defects can be improved by the random forest integration method in integrated learning bagging. Randomforest is composed of many decision trees, and there is no correlation between different decision trees. First, randomly sample the samples, train the decision tree, and then classify the nodes according to the corresponding attributes until they can no longer split the position, and build a large number of decision trees to form a forest.

3 Establishing Model

Before using the decision tree algorithm to build a tree of regional economic indicators, it is necessary to determine the target value level of the decision tree, which is also the focus of this model. First, use the entropy method to calculate the corresponding weights for the indicators of the regional economy. Select n indicators and m periods So X_{ij} represents the y-th value of the i-th index $(i = 1, 2, ..., n; j = 1, 2, ..., m)$.

3.1 Indicator Normalization

The homogeneous indicators are homogeneous. Because the measurement units of the indicators are not uniform before they are used to calculate the comprehensive indicators, they must be standardized, that is, the absolute values of the indicators are converted into relative values, and $X_{ij} = |X_{ij}|$, so as to solve the problem of homogeneity of various qualitative index values. The specific method is as follows:

$$x'_{ij} = \frac{x_{ij} - \min\{x_{1j}, \ldots, x_{nj}\}}{\max\{x_{1j}, \ldots, x_{nj}\} - \min\{x_{1j}, \ldots, x_{nj}\}} \tag{4}$$

Then X'_{ij} is the value of the j time period of the i index. For convenience, the normalized data are recorded as X_{ij}.

3.2 Calculate the Weight of Each Indicator

$$p_{ij} = \frac{x_{ij}}{\sum_{n=1}^{n} x_{ij}}, i = 1, \ldots, n, j = 1, \ldots, n \tag{5}$$

$$e_j = -k \sum_{i=1}^{n} p_{ij} \ln\left(p_{ij}\right), j = 1, \ldots, m \tag{6}$$

Calculate weights for various years:

$$w_j = \frac{d_j}{\sum_{j=1}^{m} d_j}, j = 1, 2, \ldots, m \tag{7}$$

Calculate the comprehensive score obtained by each indicator:

$$s_i = \sum_{j=1}^{m} w_j \cdot p_{ij}, i = 1, \ldots, n \tag{8}$$

In this way, the corresponding weights and scores of each index can be obtained. This score is convenient for later determination of the characteristic value of the decision tree.

3.3 Clustering for Rank

This article clusters the indicators that affect the economy. According to the "Elbow Rule" and the actual economic stage in Singapore history, the categories are finally classified into 4 categories, and then the average score of each category is calculated. It is obtained by adding the scores obtained by the entropy method of each type of index and averaging. These four scores are then ranked and divided into four intervals, and then each sample is compared to which interval the corresponding score is calculated according to the weight. Finally, each sample can obtain a corresponding rank. The rank is the target value to be evaluated by the decision tree.

3.4 Decision Tree Selection of Economic Indicators

(1) For each feature A and all possible values a, divide the data set into two subsets A = a and A! = A, and calculate the Gini index of set D:

$$\text{Gini}(D, A) = \frac{D_1}{D}\text{Gini}\left(D_1\right) + \frac{D_2}{D}\text{Gini}\left(D_2\right) \tag{9}$$

(2) Iterate through all the features A, calculate all the Gini indexes of its possible values a, select the feature corresponding to the minimum Gini index of D and the cut point as the optimal division, and divide the data into two subsets.

(3) Recursively call steps (1) (2) on the above two child nodes until the stop condition is satisfied.

4 Experiment

Before the experiment, we downloaded and organized the data from the Singapore government statistics website https://www.singstat.gov.sg/.

4.1 Data Source Analysis

The data preparation stage includes data acquisition and data preprocessing, which is the foundation of data mining. In order to make the analysis results real and effective, the data comes from the official website of the relevant region. Due to the variety of data obtained, we need to select the data appropriately according to the research goals set in advance. With reference to the existing research results, we analyzed more than 20 relevant indicators (GDP), government operating income, employment opportunities, private consumption expenditures, and output investment that affect economic development. Following the principles of scientificity, representativeness, availability, and operability of selected indicators, this article takes Singapore as an example to select 19 representative indicators, and obtains data for each quarter from 1975 to 2017 from the department of statistics of Singapore Make up the data set. The relevant indicators are shown in Table 1 below.

Table 1. The indicator system

GDP	Government spending
Aggregate demand index	Gross Fixed Capital Formation
Employee wages	Total operating surplus
Tax cuts subsidies	Residential price index
The birth rate	The company number
Gas Sales	Electricity Generation
Retail consumption index	Inbound Tourism numbers
Tourism Receipts	Air Cargo Loaded
Motor Vehicle Population	Vessel Arrivals number
Vessel Total Cargo (Thousand Tonnes)	

4.2 Processing Data Missing Values

From Singapore for the presence of some data of 19 indicators data missing value, according to the usual data before operation is to take the average and median to fill the missing value, but to fill the lack of scientific data and accuracy, so in this paper, the way is through the correlation between indicators and indicators to fill the missing value, through index correlation analysis between the first, find a missing value indicators and other indicators of relevance, through strong correlation to fill in the missing data. The correlation analysis of these 19 indicators is shown in Fig. 1 below.

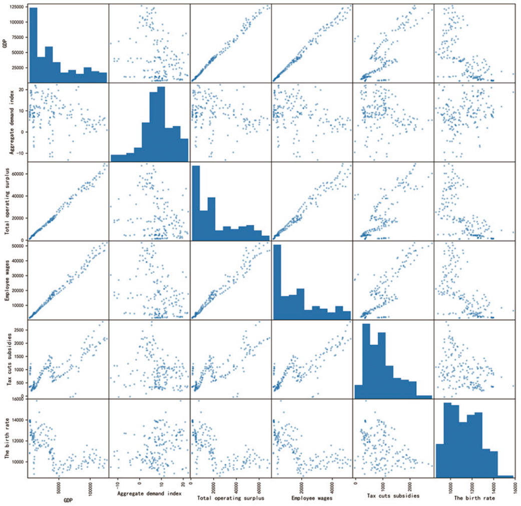


Fig. 1. Correlation scatter plots between indicators

The missing values in the 19 indicators are Total operating surplus, Employee wages, Tax cuts subsidies, The birth rate It can be seen from Fig. 1 that the linear correlation between features is very high. You can use GDP to predict Total operating surplus, use GDP to predict Employee wages, use Government spending and Total Merchandise Trade, and The company number to predict The birth rate, See Table 2 in order.

Table 2. Prediction equation

f(x)
$f(x) = 0.5284713 * x - 597.95273615$
$f(x) = 0.0365264104 * x + 289.10634$
$f(x) = 2.2291718.925 * 10^{-5} * x^2 - 1.6824388 * x + 15161.526483621026$

4.3 Ranking by Clustering and Weighting

The following is the weight of each index obtained by the entropy method, as shown in Table 3 below.

Table 3. Weight of each indicator

Index	Weight	Index	Weight
GDP	0.091025	Government spending	0.093909
Aggregate demand index	0.013786	Gross Fixed Capital Formation	0.076256
Employee wages	0.091720	Total operating surplus	0.090709
Tax cuts subsidies	0.036622	Residential price index	0.060454
The birth rate	0.035269	The company number	0.043344
Gas Sales	0.042019	Electricity Generation	0.063323
Retail consumption index	0.044275	Inbound Tourism numbers	0.069598
Tourism Receipts	0.046299	Air Cargo Loaded	0.045678
Motor Vehicle Population	0.046014	Vessel Arrivals number	0.009698

Since the data distribution is partitioned, we use the k-means algorithm to cluster the economic data sets accordingly. Before clustering, we first use the PCA dimensionality reduction algorithm to reduce the data set to 3 dimensions because there are many features and the dimensions are inconsistent, which affects the clustering too much. As shown in Fig. 2.

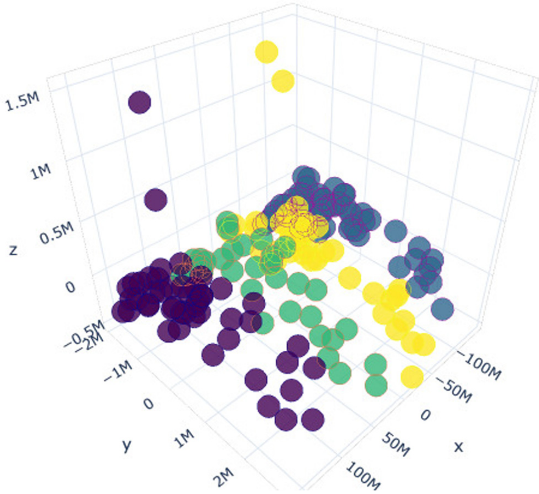


Fig. 2. Clustering graph

Based on the above clustering effects, we divide the original indicators into four broad categories. Then calculate the average for each category. The data for each category are shown in Table 4 below.

Table 4. Composite scores for four types of samples

Level	Interval score
First	$307162.4 \leq x \leq 439842.85$
Second	$222486.69 \leq x \leq 307162.4$
Third	$116022.01 \leq x \leq 222486.69$
Fourth	$x \leq 116022.01$

From the Table 4 above, we can know the score interval corresponding to each category. We use the weight score and each row of data to calculate the score and determine the level.

4.4 Generate Decision Tree

Because of the type of data in the dataset, we use the CART algorithm. The Gini index is calculated as follows:

$$GINI(D) = \sum_{i=1}^{k} p_k \cdot (1 - p_k) = 1 - \sum_{i=1}^{k} p_k^2 \tag{10}$$

The decision tree is constructed without pruning. The decision tree is mainly constructed using python machine learning third-party library sklearn. The accuracy rate obtained by the test is 84%. In order to make the model more accurate and the error smaller, a random forest is used to optimize the decision tree. The CART decision tree visualization is shown in Fig. 3.

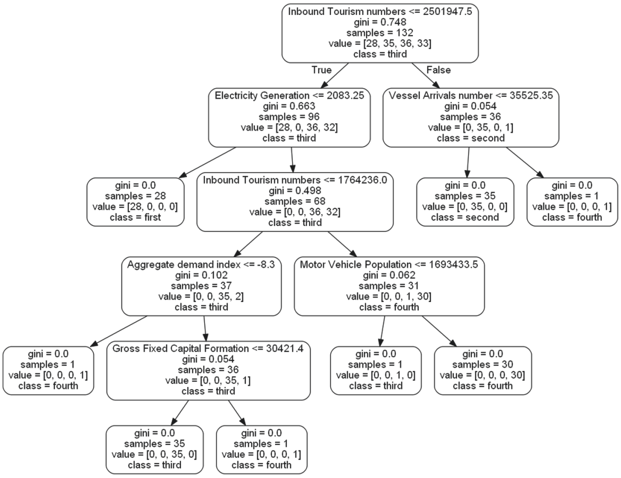

Inbound Tourism numbers <= 2501947.5 gini = 0.748 samples = 132 value = [28, 35, 36, 33] class = third
True
False
Electricity Generation <= 2083.25 gini = 0.663 samples = 96 value = [28, 0, 36, 32] class = third
Vessel Arrivals number <= 35525.35 gini = 0.054 samples = 36 value = [0, 35, 0, 1] class = second
gini = 0.0 samples = 28 value = [28, 0, 0, 0] class = first
Inbound Tourism numbers <= 1764236.0 gini = 0.498 samples = 68 value = [0, 0, 36, 32] class = third
gini = 0.0 samples = 35 value = [0, 35, 0, 0] class = second
gini = 0.0 samples = 1 value = [0, 0, 0, 1] class = fourth
Aggregate demand index <= -8.3 gini = 0.102 samples = 37 value = [0, 0, 35, 2] class = third
Motor Vehicle Population <= 1693433.5 gini = 0.062 samples = 31 value = [0, 0, 1, 30] class = fourth
gini = 0.0 samples = 1 value = [0, 0, 0, 1] class = fourth
Gross Fixed Capital Formation <= 30421.4 gini = 0.054 samples = 36 value = [0, 0, 35, 1] class = third
gini = 0.0 samples = 1 value = [0, 0, 1, 0] class = third
gini = 0.0 samples = 30 value = [0, 0, 0, 30] class = fourth
gini = 0.0 samples = 35 value = [0, 0, 35, 0] class = third
gini = 0.0 samples = 1 value = [0, 0, 0, 1] class = fourth

Fig. 3. Economic indicator decision tree

[2] here we use grid search k-fold cross-validation to adjust the parameters of the random forest parameters, and it is concluded that the effect is best when the estimator is 50. After getting the number of estimators, To further improve the accuracy, due to the limited data provided by the government, the dimension cannot reach several hundred dimensions. Here we mainly discuss the use of cross-validation to obtain the optimal *min_impurity_split* size for pruning to prevent excessive growth from causing poor generalization ability. The verification curve results are shown in Fig. 4:

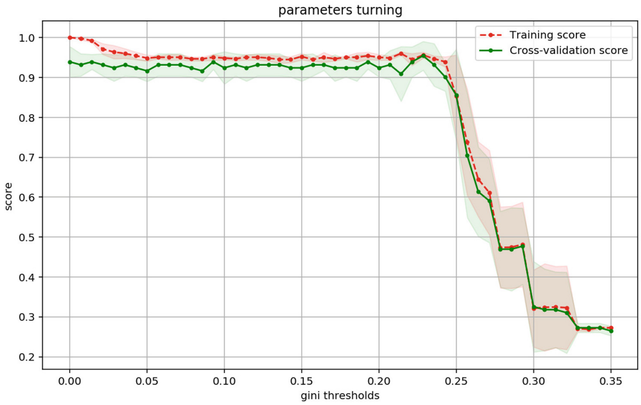


Fig. 4. Cross validation results

It can be seen from Fig. 4 that when the impurity index is 0.22857142857. Finally, the accuracy of the training set and the test set was 94.7%.

After the random forest model is obtained, each decision tree in the random forest is judged separately when a new sample is entered. The bagging set strategy is relatively simple. For classification problems, the voting method is usually used, and the most votes category or one of the categories is the final model output. The time complexity is $O(M(mnlogn))$. Some features need to be selected randomly during the calculation process, and additional time is required to process this process, so it may take more time. Where n represents n samples, m represents m features, and M represents the number of decision trees participating in the voting.

4.5 Model Importance Interpretation

This paper uses the common algorithm xgboost in boosting, and compares and verifies the feature importance obtained from the random forest algorithm in bagging. Since the GBDT algorithm only has a regression tree, it will not be discussed here. This adjusted random forest model consists of 50 lessons of decision trees. Each tree can get an impurity measure about each feature, and then the scores can be added according to the feature to get the relevant feature importance [1]. At the same time, we use the xgboost algorithm, which is also composed of 50 decision trees, to compare. After adjusting the parameters, the optimal subsample is 0.5204081. The best learning_rate is 0.3000012, the import_type is modified to weight, and the objective is modified to multisoftprob. After completion, we can get the feature importance corresponding to the two algorithms, see Fig. 5 below.

As shown in Fig. 5, assuming that the sum of the importance of all features is 1, it can be seen that in two well-known algorithms, the importance of Inbound Tourism numbers is the largest, indicating that the largest factor affecting the regional economy is Inbound Tourism numbers during the entire classification This indicator is followed by GDP. Among them, aggregate demand index and tax cuts subsidies and Air Cargo Loaded and other corresponding features account for a small proportion, indicating that in the process of economic development, the impact of these factors is small. We can draw from this model that Singapore can vigorously develop the tourism industry and prompt the corresponding GDP, which needs to be strengthened on the characteristics of lower scores.

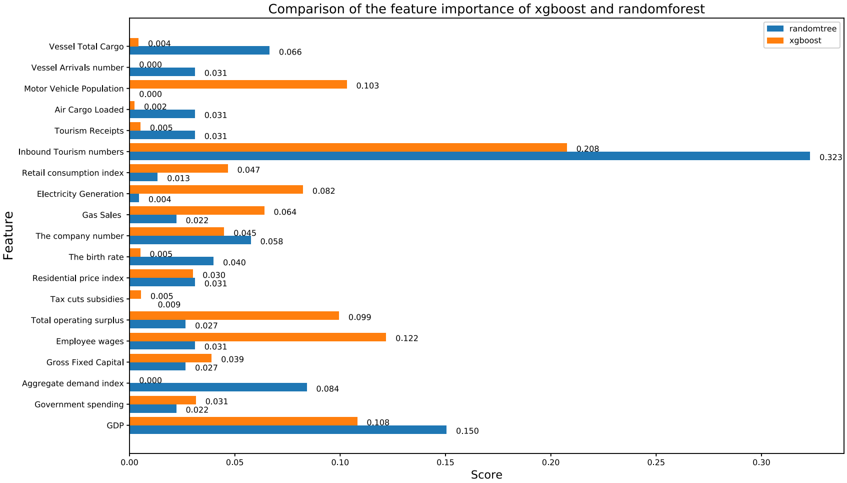


Fig. 5. Compare results

5 Conclusion

The weight of each index is determined by the entropy method, and K-means is used to cluster and divide the levels, and the target value is determined for each index. An economic index screening model based on the CART tree random forest algorithm. Through this model, it is possible to scientifically screen out those important indicators affecting the regional economy. The accuracy of the fusion of the above models is as high as 94%, and there will not be too much error when analyzing the factors affecting the regional economy. This method helps regional economic decision-makers to provide accurate positioning by providing decision support for regional economic development. The analysis of the above results shows that the Singapore government can vigorously develop the tourism industry because the importance of inbound tourism numbers is very high and the economic level development can be completed accurately.

Based on the research conclusions of this paper, the real-time analysis of larger data can be carried out based on the study of machine learning technology to screen the main factors affecting the regional economy, and an economic impact factor analysis system based on each region or country can be established. If the data supports, the number of indicators can be larger, so that the main factors affecting the regional economy can be analyzed more comprehensively and accurately.

References

1. Li, X.: Research on application of decision tree integration method in precision traffic safety publicity. In: Proceedings of the 13th China Intelligent Transportation Annual Conference, pp. 562–571. China Intelligent Transportation Association (2018)
2. Ying, Q.: Research and application of educational data mining based on decision tree technology. Zhejiang Normal University, Zhejiang (2018)
3. Lin, Z., Wang, S., Qiu, Z., Lulu, Y., Nan, M.: Application of decision tree algorithm to forecasting stock price trends. In: The 15th Network New Technology and Application Year of China Computer User Association Network Application Branch. 2011 Conference Proceedings of the 15th Annual Network New Technology and Application Conference of the Network Application Branch of China Computer Users Association in 2011, pp. 129–131. China Computer Users Association, Beijing (2011)
4. Guo, X.: Improved entropy method and its application in economic benefit evaluation. Syst. Eng. Theory Pract. (12), 99–103 (1998)
5. Chen, H., Xia, D.: Application research of data mining algorithm based on CART decision tree. Coal Technol. **30**(10), 164–166 (2011)

An Evaluation Algorithm of the Importance of Network Node Based on Community Influence

Gongzhen He(✉), Junyong Luo, and Meijuan Yin

State Key Laboratory of Mathematical Engineering and Advanced Computing,
Zhengzhou 450001, China
he_gz_study@163.com

Abstract. Identifying nodes in social networks that have great influence on information dissemination is of great significance for monitoring and guiding information dissemination. There are few methods to study the influence of communities on social networks among the existing node importance evaluation algorithms, and it is difficult to find nodes that promote information dissemination among communities. In view of this reason, this paper proposes a node importance evaluation algorithm based on community influence (abbreviated as IEBoCI algorithm), which evaluates the importance of the nodes based on the influence degree of the nodes on the communities and the ability to disseminate information the communities to which the nodes are connected. This algorithm firstly calculates the activation probability of nodes to other nodes, which is used to divide communities and evaluate influence. Secondly, the network is divided into communities based on LPA algorithm. Finally, the importance of the node is calculated by combining the influence of the community itself and the influence of the node on the community. Experiments are carried out on real social network data and compared with other community-based methods to verify the effectiveness of the algorithm.

Keywords: Complex network · Social network · Node importance · Community detection · Diffusion model

1 Introduction

With the rapid development of Internet and information technology, social networks such as Weibo, Facebook, Flikr, Twitter, etc. have developed rapidly. Social networks have become one of the main platform for human beings to spread information. Identifying nodes with great influence on information dissemination in social networks is helpful for in-depth analysis of information dissemination and evolution in social networks. Finding the guider or pusher in network public opinion is of great significance for controlling and guiding network public opinion, cracking down on network information crimes, and realizing viral marketing and word-of-mouth communication.

There are two main methods for evaluating the importance of nodes, methods based on centralities and methods based on information dissemination scale. The centrality-based method evaluates the centrality of nodes depends on the network structure, which

Y. Tan et al. (Eds.): DMBD 2020, CCIS 1234, pp. 57–70, 2020.
https://doi.org/10.1007/978-981-15-7205-0_6

represents the degree of nodes in the center of the network. These centrality methods mainly include Degree Centrality [1], Betweenness Centrality [2], Closeness Centrality [3], Eigenvector Centrality [4], etc. This kind of method is suitable for finding the important nodes in the network structure, but not for evaluating the influence of nodes.

The method based on information dissemination scale is to use information dissemination model to simulate the information dissemination process, calculate the information dissemination scale of nodes, and find out the nodes with great influence. A greedy algorithm for calculating the propagation scale of nodes was first proposed by Kempe et al. [5], which is very time consuming and only suitable for small networks. In order to reduce the computational complexity, Leskovec et al. [6] proposed CELF algorithm according to the submodules of influence diffusion, avoiding redundant calculation of activation range. References [7–11] used heuristic strategy instead of Monte Carlo simulation to estimate propagation scale for improving time efficiency. This kind of method finds influential nodes by directly measuring the information dissemination scale of nodes, but it is not suitable for evaluating the importance of nodes in the network that indirectly disseminate influence.

The methods of Cao [15], Wang [16], Shang [17], Zhang [18] and other teams assume the independence between communities. After dividing the network into communities, they find the node with the greatest local influence in each community, and then find the node with the greatest influence in the whole network. M. M. Tulu [19] et al. calculated the node's Shannon Entropy as the node's importance by using the number of nodes outside the community and the number of nodes inside the community after the community was divided. Zhao [20] measured the importance of nodes by the number of communities which the nodes connected to after dividing the network into communities. These methods either do not focus on the association between the communities, or do not consider the relationship between the nodes and the different communities, or do not consider the influence of the communities themselves.

In order to deal with the above problems, we propose a node importance evaluation algorithm based on community influence (abbreviated as IEBoCI algorithm). Its basic assumption is that the stronger the ability of the community connected by nodes to disseminate information and the greater the influence of nodes on the community, the higher the importance of nodes. The algorithm first calculates the activation probability of nodes to other nodes; Secondly, the network is divided into communities based on LPA algorithm; Thirdly, calculate the influence of each community and the influence degree of nodes on the connected communities; Finally, the importance of the node is calculated by combining the influence of the community itself and the influence of the node on the community.

2 IEBoCI Algorithm Framework

If a person has many friends in different societies in social networks, this person does not necessarily directly disseminate a large amount of important information, but he can indirectly disseminate the information in the community through contacts with other community members. From this we can see that this person has a wide influence on information dissemination and plays a more important role in the network.

We believe that the influence of nodes is related to the number and quality of communities connected by nodes based on this assumption. The more communities connected, the greater the influence of the nodes and the higher the importance. Meanwhile, the influence of the nodes is also related to the influence of the connected communities. For the same community, the influence degree of different nodes on the community is also different. If a node has less influence on a community, it is difficult for the node to influence the nodes in the community, and it is not easy to further spread information through the community. Therefore, it is necessary to comprehensively evaluate the influence of the community itself and the influence degree of nodes on the community when evaluating the importance of nodes.

3 Algorithm Steps

Social network is a complex network, which is denoted as directed network G = (N, E) in this paper, among which N is a collection of nodes in a network, E is a set of directed edges in a network. The IEBoCI algorithm proposed in this paper is based on directed network. The algorithm flow is shown in Fig. 1. The steps are as follows:

(1) calculate the activation probability of nodes activating their reachable nodes based on the information propagation model, which is used to divide communities and calculate the influence range of communities and nodes; (2) divide the network based on label propagation algorithm to obtain the community structure of the network; (3) calculate the influence range of communities according to the activation probability of the nodes; (4) calculate the number expectation of the nodes on communities activated by the nodes, and further obtaining the influence degree of the nodes on communities; (5) calculate the importance of nodes by combining the results of the third and fourth steps.

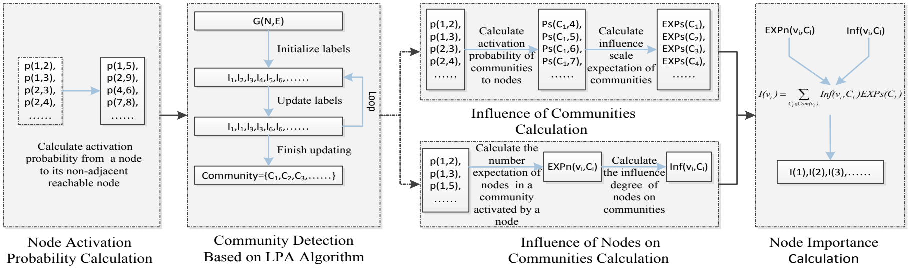

p(1,2),
p(1,3),
p(2,3),
p(2,4),
......
p(1,5),
p(2,9),
p(4,6),
p(7,8),
......
Calculate activation probability from a node to its non-adjacent reachable node
Node Activation Probability Calculation
G(N,E)
Initialize labels
$l_1,l_2,l_3,l_4,l_5,l_6,\ldots\ldots$
Update labels
$l_1,l_1,l_3,l_3,l_6,l_6,\ldots\ldots$
Loop
Finish updating
Community={C1,C2,C3,......}
Community Detection Based on LPA Algorithm
p(1,2),
p(1,3),
p(2,3),
p(2,4),
......
Calculate activation probability of communities to nodes
Ps(C1,4),
Ps(C1,5),
Ps(C1,6),
Ps(C1,7),
......
Calculate influence scale expectation of communities
EXPs(C1),
EXPs(C2),
EXPs(C3),
EXPs(C4),
......
Influence of Communities Calculation
p(1,2),
p(1,3),
p(1,5),
......
Calculate the number expectation of nodes in a community activated by a node
EXPn(vi,Cl)
Calculate the influence degree of nodes on communities
Inf(vi,Cl)
Influence of Nodes on Communities Calculation
EXPn(vi,Cl)
Inf(vi,Cl)
$I(v_i)=\sum_{C_l\in Com(v_i)} Inf(v_i,C_l)EXPs(C_l)$
I(1),I(2),I(3),......
Node Importance Calculation

Fig. 1. Algorithm flow chart

3.1 Node Activation Probability Calculation

This paper calculates the information dissemination scale of nodes and communities based on independent cascade model (IC model). IC model is a probability model, and there is a probability $p(v_i, v_j) \in [0, 1]$ for all neighboring nodes v_i and v_j in network G. A value between 0 and 1 is randomly assigned, which indicates the probability that the active node v_i successfully directly activates the neighbor node v_j. For non-adjacent nodes v_i and v_j, if v_i to v_j are unreachable, the probability of v_i activating v_j is 0, that is $p(v_i, v_j) = 0$. If v_i is reachable to v_j, the probability that node v_i activates node v_j along the path is the product of one node directly activating another node on each side of the path [22]. If there are m paths between v_i and v_j, one of the paths is $Path(v_i, v_j)_x = <v_j = v_1, v_2, \ldots, v_j = v_k>$, the probability $Pp(v_i, v_j)_x$ that node v_i activates node v_j along this path is calculated as follows:

$$Pp(v_i, v_j)_x = \prod_{u=1}^{k-1} p(v_u, v_{u+1}) \tag{1}$$

Where $Pp(v_i, v_j)_x$ is the probability that v_i activates v_j through $Path(v_i, v_j)_x$ and $p(v_u, v_{u+1})$ is the probability that node v_u activates neighbor node v_{u+1}. If the activation probabilities of different paths are different, the maximum probability is taken as the probability $p(v_i, v_j)$ for node v_i to activate non-adjacent reachable node v_j.

3.2 Community Detection

We divide network into communities based on label propagation algorithm (LPA algorithm) to obtain a collection of communities on the network. LPA algorithm is applicable to undirected and unweighted networks. The social network constructed in this paper is a directed network. Therefore, when calculating the labels to be updated of node v_j, only the in-neighbors of node v_j are calculated, and the labels with the highest activation probability among the neighbors are counted. The steps are as follows:

step1: Label initialization: each node in the network is randomly assigned a unique label l, which represents the community in which the node is located
step2: Determining the node order of the asynchronous updating labels: calculate the degree of the nodes, and arrange the node order of the asynchronous updating labels from large to small according to the degree of the nodes;
step3: Updating the labels of nodes: according to the node order of updating the labels, the labels of nodes are updated one by one, and the label of node v_j is updated to the label with the maximum sum of activation probabilities in its in-neighbor nodes. The label updating formula is as follows:

$$l_{v_j} = \arg\max_{l} \sum_{i \in IN(v_j)} p(v_i, v_j)\delta(l_{v_i}, l) \tag{2}$$

l_{v_j} represents the label of node v_j to be updated, l_{v_i} represents the label of node v_i, $IN(v_j)$ represents the set of nodes with out-edges to node v_j, $p(v_i, v_j)$ represents the activation probability from node v_i to node v_j, and $\delta(l_i, l)$ is a Kronecker function.

When there is more than one label with the maximum sum of the calculated activation probabilities, one label is randomly selected from them as the new label of the node.

Step4: Termination judgment: it is judged whether the labels of all nodes in the network are the labels with the largest sum of activation probabilities among neighbor nodes. If not, step3 is repeatedly executed, if so, calculation is terminated, and nodes with the same label belong to the same community.

3.3 Evaluation of Influence of Communities

In this paper, the number expectation of network nodes activated by a community is taken as the influence of the community. The steps are as follows: firstly, the activation probability between nodes calculated by 3.1 is used to calculate the joint activation probability of all nodes in the community to nodes in the network; then calculate the number expectation of nodes activated by the community according to the joint activation probability to obtain the influence of the community.

In the independent cascade model, whether a node activates another node and whether other nodes activate the node are independent events, so the joint activation probability Ps(C_l, v_j) [23] of the community C_l to a node v_j in the network is calculated according to the probability multiplication of the independent events, and the calculation formula is:

$$Ps(C_l, v_j) = 1 - \prod_{v_i \in C_l} (1 - p(v_i, v_j)) \tag{3}$$

With the joint activation probability Ps(Cl, vj), the influence scale expectation of the community EXPs(Cl) is calculated as follows:

$$EXPs(C_l) = \sum_{v_j \in N} Ps(C_l, v_j) \tag{4}$$

Where N is the set of all nodes in the network.

3.4 Evaluation of Influence of Nodes on Communities

The number expectation of nodes in a community activated by a node indicates how many nodes in the community a node can successfully activate. The number of nodes in the community indicates the total scale of the community. The greater the proportion of nodes in the community that a node can activate in all nodes of the community, the greater the influence of the node on the community. Therefore, this paper regards the ratio of number expectation of nodes in a community activated by a node to the number of nodes in the community as the influence degree of the node on the community.

Limiting the range of nodes activated by node v_i in the community C_l, the number expectation of nodes $EXPn(v_i, C_l)$ in the community C_l activated by node v_i is obtained, which is equal to the sum of the probabilities of each node in the community Cl being successfully activated by node v_i, and the calculation formula is as follows:

$$EXPn(v_i, C_l) = \sum_{v_j \in C_l} p(v_i, v_j) \tag{5}$$

Where $\mathrm{C_l}$ represents the set of all nodes in the community $\mathrm{C_l}$.

$EXPn(v_i, C_l)$ represents the number of nodes that node $\mathrm{v_i}$ can activate in community Cl. The ratio of this expectation to the total number of nodes $\mathrm{n(C_l)}$ in community $\mathrm{C_l}$ is the influence degree $Inf(v_i, C_l)$ of node $\mathrm{v_i}$ on community $\mathrm{v_i}$. The formula is:

$$Inf(v_i, C_l) = \frac{EXPn(v_i, C_l)}{n(C_l)} \tag{6}$$

3.5 Node Importance Evaluation

Node $\mathrm{v_i}$ can use influence of communities directly connected by node $\mathrm{v_i}$ to spread influence indirectly. We calculate the sum of the influence of communities that node $\mathrm{v_i}$ can indirectly use, and get the importance $\mathrm{I(v_i)}$ of node $\mathrm{V_i}$. The importance $\mathrm{I(v_i)}$ of the node $\mathrm{v_i}$ is calculated by the following formula using influence of communities and the influence of a node on community:

$$I(v_i) = \sum_{C_l \in Com(v_i)} Inf(v_i, C_l) EXPs(C_l) \tag{7}$$

Where $\mathrm{Com(v_i)}$ represents the set of communities in which node $\mathrm{v_i}$ and its out-neighbors are located.

4 Experimental Results and Discussions

4.1 Experimental Data and Initial Setup

The experimental data used in this paper are commonly used public social network data sets, which are downloaded from Internet. The name, data scale and description of the network are shown in Table 1.

Table 1. Basic information of datasets

Data set	Number of nodes	Number of edges	Network density	Average out degree	Minimum out degree	Maximum out degree	Description
Facebook[a]	4039	176468	0.0108200	43.691	1	1045	Friends on Facebook
E-Mail[b]	1866	5517	0.0015853	2.9566	0	330	Communication between users

Note: [a]https://snap.stanford.edu/data/ego-Facebook.html
[b]http://konect.uni-koblenz.de/networks/dnc-temporalGraph

The nodes in Facebook are friends relationship, and the formed network is an undirected network. In this experiment, each edge is converted into two directed edges to convert undirected network into directed network. The nodes in the mail are communication relationship. Each communication forms a directed edge from one node to another node.

In order to simulate the dissemination of information on the network, this paper uses node v_i as the initial activation node, and the scale of the nodes that can be affected by node v_i as a measure of the node's information dissemination capability, which is referred to as the Influence scale. In order to simulate the influence propagation process of nodes and calculate the influence scale of nodes, the commonly used independent cascade model (IC model) [21] is adopted in this paper.

The activation probability p between nodes is randomly assigned a value between 0 and 1 when constructing a network using data sets. Due to the randomness of the independent cascade model, the results may be different when calculating the influence scale of nodes. To sum up, each node is taken as the initial activated node to calculate the influence scale for 50 times when calculating the influence scale of nodes, and then the arithmetic average value is taken as the final result.

4.2 The Division and Influence of Communities

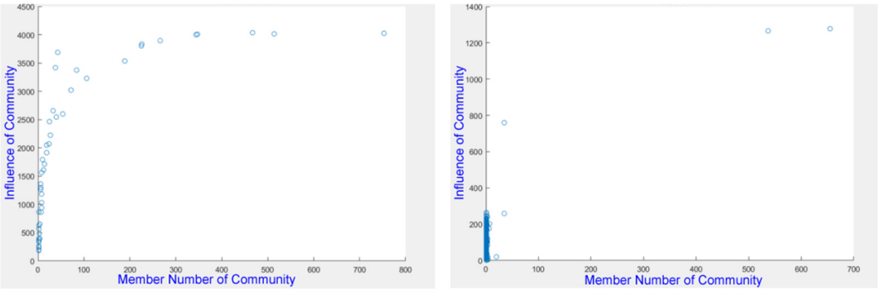
The two data sets are divided into communities after the network construction is completed, according to the activation probability between nodes by using the LPA algorithm improved previously. The division results are shown in Tables 2 and 3.

Table 2. Communities of Facebook data sets

Community size	Number of communities	Community size	Number of communities
2	10	40	1
3	1	43	1
4	2	54	1
6	3	72	1
7	1	84	1
8	4	106	1
10	1	189	1
12	1	225	1
14	1	226	1
19	2	266	1
24	1	344	1
25	1	347	1
27	1	467	1
33	1	514	1
38	1	753	1
Total number of communities		46	

Table 3. Communities of mail data sets

Community size	Number of communities
1	490
2	27
3	5
4	3
6	1
7	1
20	1
35	2
537	1
655	1
Total number of communities	532



(a) Influence of Communities in Facebook Data Set

(b) Influence of Communities in E-Mail Data Set

Fig. 2. Influence of communities

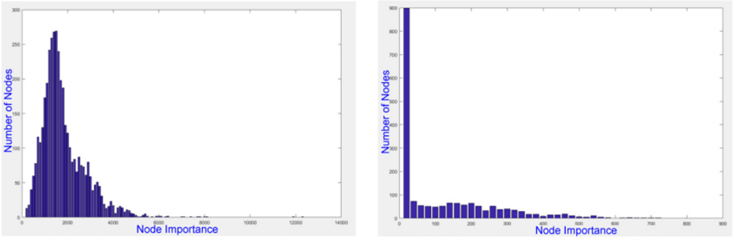
According to the results of community detection, it can be seen that the structural characteristics of Facebook and email are quite different: Facebook has 4039 nodes and 46 communities are divided, with a small number of communities and a large scale of communities, which indicates that the network is relatively close. There are 1866 nodes in the mail network, 532 of which are divided into communities. The number of communities is large, but the number of large-scale communities is small. More communities are 1 node and 2 nodes, which shows that the network is sparse.

The influence of communities is calculated after the community detection is completed, according to the method proposed in this paper, and the calculation results are shown in Fig. 2. Figure 2. (a) shows the influence of communities in Facebook data. The number of communities is small, and the size of communities (the number of members of communities) varies greatly. From the overall trend, the larger the size of communities, the greater the influence of communities. When the influence of the community reaches a certain degree (the number of nodes affected reaches more than 90% of the total number of nodes), the increase in influence becomes less and less obvious, which is consistent with the reality. Figure 2. (b) shows the influence of communities in E-Mail data. Generally speaking, there is also a trend that “the larger the community size, the greater the influence of the community”. As there are a large number of 1-node and 2-node communities in the network, the data in the lower left corner of the image is relatively dense, and the influence of communities is not necessarily the same under the same scale.

4.3 The Importance of Node

According to the method proposed in this paper, the importance I of all nodes in data sets and the Influence scale of nodes are calculated, and compared with a method of indirectly measuring the importance of nodes through communities (the number of directly connected communities V-community [20]). The Degreeout and Betweenness are analyzed as statistical data.

Distribution of Node Importance. After calculating the importance of nodes, numbers of nodes with different values of importance were counted in Facebook and E-Mail data sets. The number of nodes was counted in Facebook data set according to the importance with 100 as an interval, and the number of nodes was counted in E-Mail data set according to the importance with 20 as an interval. The distribution of statistical results is shown in Fig. 3. The distribution in the two data sets is different. The distribution in Facebook is positively skew distribution and the distribution in mail is power law distribution. The difference between the two results lies in the different characteristics of the two data sets. The network in Facebook data is a directed network transformed from an undirected network, and each node has an out-degree greater than 0, so each node can transmit information to other nodes. In the mail data, the directed network is constructed according to the communication relationship. There are a large number of nodes in the data set that receive mail but do not send mail, which have an out-degree of 0 and do not carry out information dissemination to the outside. Therefore, a large number of nodes with an out-degree of 0 result in a large number of nodes with low importance in the data statistics. So the distribution of importance presents a power law distribution.

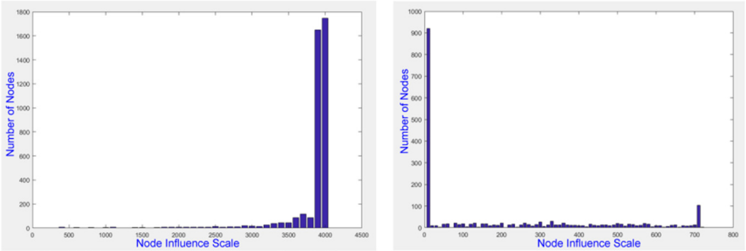


(a) The Node Importance Distribution in Facebook Data Set

(b) The Node Importance Distribution in E-Mail Data Set

Fig. 3. Node importance statistics

Comparison of Node Importance and Number of Directly Connected Communities. The number distribution statistics of the Influence scale of nodes in the two data sets are shown in Fig. 4.



(a) The Node Influence Scale Distribution in Facebook Data Set

(b) The Node Influence Scale Distribution in E-Mail Data Set

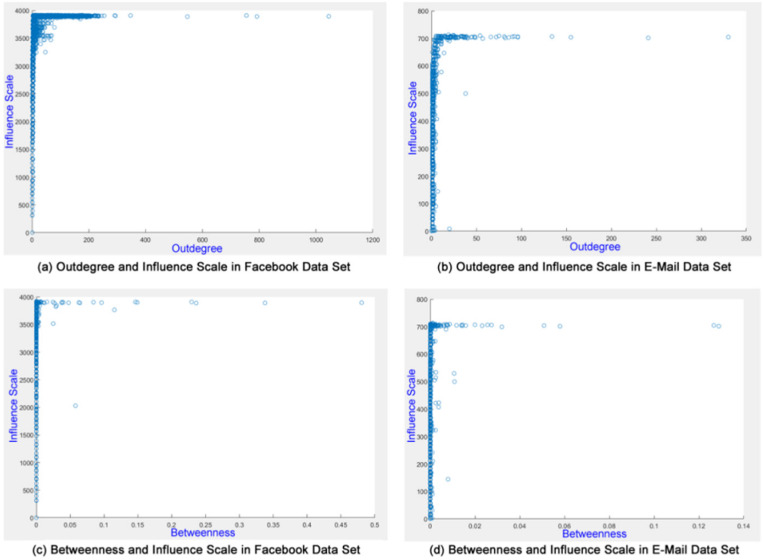
Fig. 4. Node influence scale statistics

In Fig. 4, (a) Facebook data counts the number of nodes according to the node Influence scale with 100 as an interval, and (b) E-Mail data counts the number of nodes according to the node Influence scale with 20 as an interval.

Most of the nodes in the Facebook dataset have a large scale of influence, with 1746 nodes in the [3900, 4000) interval and 1649 nodes in the [3800, 3900) interval. Facebook dataset has 4,039 nodes, of which three-quarters can affect 95% of the network. This result is also related to the close connection of nodes in the data set. The density of the network is high, the average outdegree of nodes is also large, and the influence spread range of most nodes is large.

The influence scale of most nodes in E-Mail data set is very small, 921 nodes are in [0, 20) interval. This result is related to the fact that nodes in the data set are not closely related. Different nodes have different impact sizes. There are 1866 nodes in the data set, of which 800 nodes have an output of 0. These nodes cannot transmit information outward, so nodes with low Influence scale account for the majority.

The relationship between node outdegree, betweenness and node influence scale is shown in Fig. 5. In the chart, the X axis represents the node outdegree and betweenness for the corresponding data set, and the Y axis represents the node Influence scale. In Fig. 5. (a) (b), it can be seen that there is almost no correlation between the node influence scale and the node output. In Fig. 5. (c) (d), it can also be seen that there is almost no correlation between node influence scale and node betweenness.



(a) Outdegree and Influence Scale in Facebook Data Set

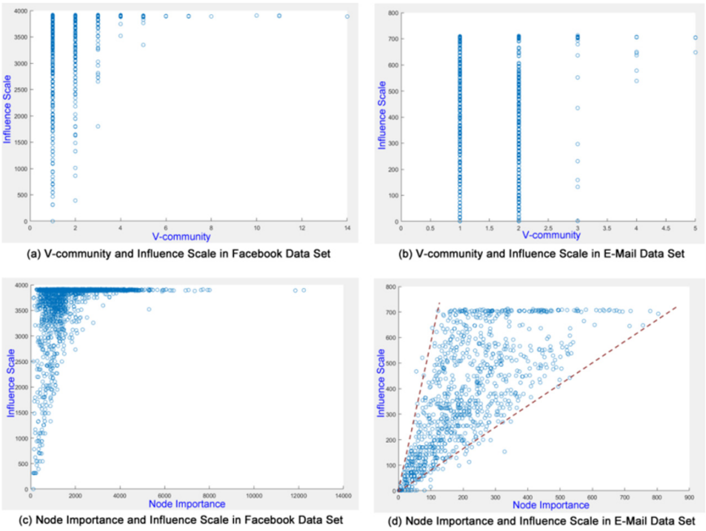
(b) Outdegree and Influence Scale in E-Mail Data Set

(c) Betweenness and Influence Scale in Facebook Data Set

(d) Betweenness and Influence Scale in E-Mail Data Set

Fig. 5. Relationship between outdegree, betweenness and node influence scale

The relationship between number of directly connected communitie (V-community), node importance and node influence scale is shown in Fig. 6. In the chart, theX axis represents V-community and node importance for the corresponding data set, and the Y axis represents the node Influence scale. In Fig. 6. (a) (b), it can be seen that there is a certain correlation between the influence scale of nodes and the number of communities directly connected by nodes. Nodes with a large number of directly connected communities have a larger influence scale, but the number of communities connected by nodes with a larger influence scale is not necessarily large. It can be seen from Fig. 6. (c) (d) that there is a strong correlation between node Influence scale and node importance. The image in Fig. 6. (c) has a larger value range of X axis and a larger image density on the left. for convenience of observation, the distribution image in the range of importance 0 to 5000 is captured, as shown in Fig. 7. As can be seen from Fig. 7, nodes with low importance may have a higher Influence scale, nodes with high importance have a higher Influence scale. In Fig. 6. (d), the correlation between node influence scale and node importance is more obvious, and the image is basically scattered between two oblique lines passing through the origin (oblique lines have been marked in the figure). There is a strong correlation between node influence scale and node importance, which shows that the nodes with high importance we find through this method have high Influence scale.



(a) V-community and Influence Scale in Facebook Data Set

(b) V-community and Influence Scale in E-Mail Data Set

(c) Node Importance and Influence Scale in Facebook Data Set

(d) Node Importance and Influence Scale in E-Mail Data Set

Fig. 6. Relationship between V-community, node importance and node influence scale

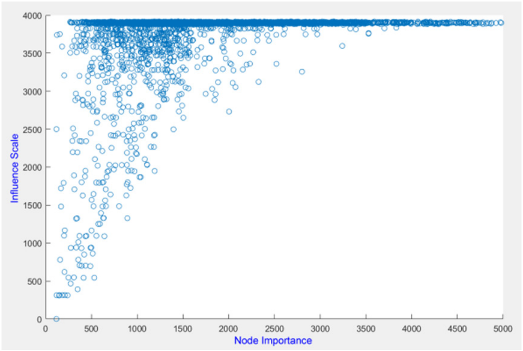


Fig. 7. Node importance and node influence scale in Facebook dataset

The data of the top ten nodes in Facebook data set are shown in Table 4, and the data of the top ten nodes in mail data set are shown in Table 5.

It can be observed that for nodes with high I, the Influence scale is very high from Table 4 and Table 5; for nodes with high I, the $\text{Degree}_{\text{out}}$ is not necessarily high, and some are even very low. I high node's Betweenness is not necessarily high, some even very low; for nodes with high I, the V-community is not necessarily high, and some are even very low.

In conclusion, the influence of nodes in the network is not related to structural features such as node degree and betweenness. The method in this paper evaluates the importance of nodes based on community influence. Nodes with higher importance

Table 4. Top ten nodes in Facebook data set in node importance

Node number	$\text{Degree}_{\text{out}}$	Betweenness	V-community	I	Influence scale
107	1045	0.480518	11	12227.468	3874.939
3437	547	0.236115	14	11849.017	3622.140
563	91	0.062780	7	8008.597	3791.743
1593	32	0.000553	6	7922.979	3495.210
0	347	0.146305	11	7714.364	3700.363
1173	115	0.000942	6	7627.214	3602.769
606	91	0.000997	5	7343.892	3440.250
1687	43	0.000907	5	7064.857	3796.674
1684	792	0.337797	8	6935.021	3478.205
428	115	0.064309	7	6397.974	3582.737

Table 5. Top ten nodes of node importance in E-Mail data set

Node number	$\text{Degree}_{\text{out}}$	Betweenness	V-community	I	Influence scale
1957	3	0	3	803.4425465	693.26
1159	155	0.050850	5	778.5791855	703.88
1312	6	0	5	756.1487419	647.72
993	44	0.015875	3	718.0338654	704.08
1882	4	0	4	714.651491	645.12
1669	241	0.128871	3	693.0661212	701.62
869	36	0.003789	3	683.0508954	702.42
1	96	0.025603	3	674.3651542	706.34
1618	17	0	5	667.307672	704.86
585	75	0.008709	2	653.1068147	708.86

have greater influence, which can promote information dissemination, and can find some nodes with less prominent structural characteristics but larger actual influence.

5 Conclusion

Identifying nodes that have greater influence on the dissemination of information in social networks is a hot research field in social networks. However, there are few methods in the existing algorithms for evaluating node importance to study the influence of communities on the dissemination of information in social networks. In this paper, a node importance evaluation algorithm based on community influence is proposed. After the network is divided into communities, the influence of communities and the influence of nodes on communities are evaluated. Finally, the importance of nodes is comprehensively evaluated by combining both. The experimental results show that the nodes with high importance evaluated by the algorithm have high influence in the network, which can promote the dissemination of information, and can find some nodes with high influence but not prominent structural characteristics.

For the next step, we plan to introduce content features and user behavior features in social networks to integrate the importance of computing nodes.

References

1. Pastor-Satorras, R., Vespignani, A.: Epidemic spreading in scale-free networks. Phys. Rev. Lett. **86**(14), 3200–3203 (2001)
2. Freeman, L.C.: A set of measures of centrality based on betweenness. Sociometry **40**(1), 35–41 (1977)
3. Sabidussi, G.: The centrality index of a graph. Psychometrika **31**(4), 581–603 (1966)
4. Bonacich, P.F.: Factoring and weighting approaches to status scores and clique identification. J. Math. Sociol. **2**(1), 113–120 (1972)
5. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington DC, USA, pp. 137–146 (2003)
6. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Costeffective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, USA, pp. 420–429 (2007)
7. Kimura, M., Saito, K.: Tractable models for information diffusion in social networks. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 259–271. Springer, Heidelberg (2006). https://doi.org/10.1007/11871637_27
8. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, pp. 199–208 (2009)
9. Chen, W., Yuan, Y., Zhang, L.: Scalable influence maximization in social networks under the linear threshold model. In: Proceedings of IEEE 10th International Conference on Data Mining (ICDM 2010), Sydney, Australia, pp. 88–97 (2010)
10. Goyal, A., Lu, W., Lakshmanan, L.V.: SIMPATH: an efficient algorithm for influence maximization under the linear threshold model. In: 2011 IEEE 11th International Conference on Data Mining (ICDM 2011), Vancouver, Canada, pp. 211–220 (2011)

11. Kimura, M., Saito, K., Nakano, R., et al.: Extracting influential nodes on a social network for information diffusion. Data Min. Knowl. Discov. **20**(1), 70–97 (2010)
12. Wu, X., Liu, Z.: How community structure influences epidemic spread in social networks. Phys. A: Stat. Mech. Appl. **387**(2–3), 623–630 (2008)
13. Huang, W., Li, C.: Epidemic spreading in scale-free networks with community structure. J. Stat. Mech: Theory Exp. **2007**(01), P01014–P01014 (2007)
14. Chu, X., Guan, J., Zhang, Z., et al.: Epidemic spreading in weighted scale-free networks with community structure. J. Stat. Mech.: Theory Exp. **2009**(7), P07043 (18 pp.) (2009)
15. Cao, T., Wu, X., Wang, S., Hu, X.: OASNET: an optimal allocation approach to influence maximization in modular social networks. In: Proceedings of the ACM Symposium on Applied Computing, Sierre, Switzerland, pp. 1088–1094 (2010)
16. Wang, Y., Cong, G., Song, G., Xie, K.: Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, pp. 1039–1048 (2010)
17. Shang, J., Zhou, S., Li, X., et al.: CoFIM: a community-based framework for influence maximization on large-scale networks. Knowl.-Based Syst. **117**(FEB), 88–100 (2017)
18. Zhang, X., Zhu, J., Wang, Q., et al.: Identifying influential nodes in complex networks with community structure. Knowl. Based Syst. **42**, 74–84 (2013)
19. Tulu, M.M., Hou, R., Younas, T.: Identifying influential nodes based on community structure to speed up the dissemination of information in complex network. IEEE Access **6**, 7390–7401 (2018)
20. Zhao, Z.Y., Yu, H., Zhu, Z.L., et al.: Identifying influential spreaders based on network community structure. Chin. J. Comput. **37**, 753–766 (2014)
21. Goldenberg, J., Libai, B., Muller, E.: Using complex systems analysis to advance marketing theory development: modeling heterogeneity effects on new product growth through stochastic cellular automata. Acad. Market. Sci. Rev. **9**(3), 1–18 (2001)
22. Huang, H., Shen, H., Meng, Z.: Community-based influence maximization in attributed networks. Appl. Intell. **50**(2), 354–364 (2020)
23. Li, J., Wang, X., Deng, K., et al.: Most influential community search over large social networks. In: 2017 IEEE 33rd International Conference on Data Engineering (ICDE), pp. 871–882. IEEE Computer Society (2017)

PCA Based Kernel Initialization for Convolutional Neural Networks

Yifeng Wang[1], Yuxi Rong[1], Hongyue Pan[1], Ke Liu[1], Yang Hu[2], Fangmin Wu[2], Wei Peng[2], Xingsi Xue[3], and Junfeng Chen[4(✉)]

[1] China Three Gorges Corporation, Beijing 100038, China
{wang_yifeng1,rong_yuxi,pan_hongyue,liu_ke1}@ctg.com.cn
[2] Hangzhou HuaNeng Engineering Safety Technology Co., Ltd.,
Hangzhou 311121, China
mark_hnsafet@qq.com, 329537060@qq.com, 12492687@qq.com
[3] Fujian Key Lab for Automotive Electronics and Electric Drive,
Fujian University of Technology, Fuzhou 350118, Fujian, China
jack8375@gmail.com
[4] College of IoT Engineering, Hohai University, Changzhou 213022, Jiangsu, China
chen-1997@163.com

Abstract. The initialization of Convolutional Neural Networks (CNNs) is about providing reasonable initial values for the convolution kernels and the fully connected layers. In this paper, we proposed a convolution kernel initialization method based on the two-dimensional principal component analysis (2DPCA), in which a parametric equalization normalization method is used to adjust the scale between each neuron weight. After that the weight initial value can be adaptively adjusted according to different data samples. This method enables each neuron to fully back-propagate errors and accelerate network model training. Finally, a network model was built and experiments were performed using *Tanh* and *ReLU* activation functions. The experimental results verify the effectiveness of the proposed method through the distribution of histograms and the curve comparison diagrams of model training.

Keywords: Convolutional neural networks · Convolution kernel initialization · PCA · Parametric equalization normalization

1 Introduction

The Convolutional Neural Networks (CNNs) [2], as representative deep learning models, have a remarkable ability to extract features directly from the original images and recognize the rules of these visual images with minimal preprocessing. The most common form of the CNNs architecture stacks a number of convolutional and pooling layers optionally followed by fully connected layers. Most notable among these is the convolution kernel which is a small trainable matrix used for features detection. The initialization of CNNs is about providing reasonable initial values for the convolution kernels and the fully connected layers.

Y. Tan et al. (Eds.): DMBD 2020, CCIS 1234, pp. 71–82, 2020.
https://doi.org/10.1007/978-981-15-7205-0_7

The kernel initialization of the CNNs is an issue worthy of discussion. For simple Neural Networks (NN), random initialization would be a good choice. G. Thimm and E. Fiesler designed experiments to test the random weight initialization methods on the multilayer perceptron and the high order networks [8]. He et al. found that the rectifying activation unit is very important for the neural network, and proposed a Parametric Rectified Linear Unit (PReLU) to generalize the traditional rectified units. It is well to be reminded that they derived a robust initialization method, particularly considering the rectifier nonlinearities [1]. Sun et al. proposed Multi-layer Maxout Networks (MMN) with multi-layer which can train active function, and deduced a new initialization method dedicated to the activation of MMN. The method can reduce the movement of internal covariates when the signal propagates through the layer [6]. A. Pacheco determined the input weights and bias for the Extreme Learning Machine (ELM) by using the Restricted Boltzmann Machine (RBM), named as RBM-ELM [4]. Similarly, Zhang and Ji also constructed a RBM model to pre-train convolution kernels. The trained weight matrix is transformed to initialize the convolution kernel parameters of the CNNs [11]. Liu et al. proposed an image extraction algorithm by mixing the AutoEncoder and the CNNs. They utilized the AutoEncoder to train the basic elements of image and initialized the convolution kernel of the CNNs [3]. Yang et al. used sparse coding to extract the convolution kernel for initialization, which can shorten the training time and raise the recognition rate [9]. In target super resolution, Li et al. proposed a Multi-channel Convolution image Super-Resolution (MCSR) algorithm, which used a residual CNN based on sparse coding and an MSRA initialization method to accelerate model training the convergence [10]. M. S. Seyfioğlu compared two NN initialization methods, unsupervised pre-training and transfer-learning, in training the deep NN on small data sets [5]. Tang et al. employed k-means unsupervised feature learning as the pre-training process, instead of the traditional random initialization weights [7]. So far, there are no unified understanding and development methods for the initialization problem of the CNNs. Moreover, the current initialization methods gave no considerations to the sample information and cannot automatically adapt to variation of the samples. In this paper, we attempt to employ the Principal Component Analysis (PCA) into kernel initialization and propose a parametric equalization normalization to adjust the scale among the neuron weights.

The rest of the article is organized as follows. Section 2 reviews the methods of the convolution kernel initialization. Section 3 proposed the PCA-based convolution kernel initialization with balanced normalization. Section 4 presents the experimental configuration and results; finally, the conclusions are presented in Sect. 5.

2 Convolution Kernel Initialization

There are mainly three initialization methods: random initialization, Xavier initialization and MSRA Initialization.

2.1 Random Initialization

The random initialization method generally refers to the normal distribution randomization method. It defines a random variable x, which obeys a probability distribution $f(x)$ with mean μ and standard deviation σ. We have the following probability density function.

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{1}$$

The random variable distribution obeys the normal distribution, referred to as $f(x) \sim N(\mu, \sigma^2)$ are set as different values. The mean μ affects the symmetry center of the curve, and the standard deviation σ influences the smoothness of the curve. The larger the μ is, the smoother the curve will be. The random initialization method initializes each parameter of the convolution kernel according to the Gaussian probability distribution.

2.2 Xavier Initialization

Xavier Glorot et al. [12] proposed the Xavier initialization method whose core idea is to keep the variance of input and output consistent and prevent all output values from going to 0. The literature derivation gives the specific form of Xavier initialization:

$$W \sim U\left(-\frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}\right) \tag{2}$$

where n_j is the total number of input neurons for each feature map of the layer, and n_{j+1} is the total number of output neurons. The U denotes a uniform distribution within this range. Each weight parameter is randomly initialized to random value which obeys this uniformly distribution. The parameters of the method only depend on the number of input and output neurons, do not need to manually set parameters, and are completely adaptive to the size of the network model itself, and are more stable and easy to use than the random initialization method.

2.3 MSRA Initialization

The MSRA initialization method is an improved method for Xavier. It is particularly applicable to the most popular *Relu* function. Its specific form is shown in (3). It is a normal distribution with a mean 0 and a variance $2/n$.

$$W \sim G\left[0, \sqrt{\frac{2}{n}}\right] \tag{3}$$

where n is the number of inputs to the layer, and G denotes that obeys a normal distribution. The parameters of this method are only dependent on the number of neurons of one input, less than the two parameters of Xavier, and are particularly suitable for *Relu* functions.

3 PCA-Based Convolution Kernel Initialization with Balanced Normalization

In this section, we employ the Two-Dimensional Principal Component Analysis (2DPCA) to extract the feature vectors of the image and introduce an equalization normalization method to adjust the scale of the weight between layers.

3.1 Two-Dimensional Principal Component Analysis

Suppose there is an image sample set $X = \{X_1, X_2, \cdots, X_N\}$, the number of sample sets is N, and the dimension is $m \times n$. Projecting images in both row and column directions, so that the variance remaining in the subspace is the largest. The fact that 2DPCA uses orthogonal transformation to eliminate the correlation between the original vectors should be paid attention too. The two principal components obtained are linearly independent. Therefore, the principal component may represent higher information than the original data. The specific 2DPCA calculation steps are as follows:

(1) Calculate the average of all image samples: $\overline{X}=\frac{1}{N}\sum_{i=1}^{N} X_i$;

(2) Calculate the covariance matrix: $G_t=\frac{1}{N}\sum\limits_{j=1} \left(X_i - \overline{X}\right)^T \left(X_i - \overline{X}\right)$;

(3) The eigenvalue decomposition of the covariance matrix G_t is performed by using the Jacobian method to obtain the corresponding eigenvalue $\lambda_1, \lambda_2, \cdots, \lambda_d$ and the eigenvector corresponding to each eigenvalue. The eigenvector corresponding to the largest k eigenvalues is selected to compose the projection matrix $U= [u_1, u_2, \cdots, u_k] \in R^{n \times k}$;

(4) Do feature extraction to every sample data by column: $F_j = X_j \cdot U \in R^{m \times k}$, what is obtained is the feature of X_j. The original two-dimensional $m \times n$ size image is now reduced to $m \times k$, that is, the number of bits of the matrix column vectoris compressed after the feature extraction, and the number of row is not changed;

(5) After the above process, a new sample F_j, $j = 1, 2, \cdots, N$ is obtained. On the new sample, repeat step (1) (2) construct a new covariance matrix G_t^*: $G_t^* = \frac{1}{N}\sum\limits_{i=1}^{N} \left(F_i - \overline{F}\right)^T \left(F_i - \overline{F}\right)$;

(6) Similarly, repeat step (3) to take the feature vector corresponding to the largest d eigenvalues to obtain the projection matrix in the row direction $V= [v_1, v_2, \cdots, v_d]$.

Since the variance of each principal component is gradually reduced, the key information included is also decremented. So generally, the contribution rate can indicate the amount of information occupied by a principal component. Specifically, it refers to the proportion of the cumulative value of the total variance of a principal component, also, the proportion of the sum of a feature value to the sum of all feature values, which is:

$$\eta = \frac{\lambda_i}{\sum\limits_{i=1}^{d} \lambda_i} \tag{4}$$

3.2 2DPCA Initialization

The 2DPCA method is used to initialize each convolution kernel in the network. The initialization is based on training sample data. The size of the convolution kernel is $R_k \times R_k$. The size of the input map for each layer is set to $C_k \times C_k$. There are a total of D_k maps. The process is as follows.

(1) Manually select one picture for each category from all the data. That is, a total of n pictures can be selected. Each picture can correspond to a category of its own.

(2) Count the number of neurons input and output for each convolutional layer, and use the Xavier initialization method to initialize all the parameter weights $W_{k,i,j}$ for each convolutional layer in turn. The $W_{k,i,j}$ denotes the weight of the j-th window of the i-th convolution kernel in the k-th layer, and all offsets are initialized to 0.

(3) Do forward-propagating to each picture, and obtain the corresponding feature map $C_{k,i,z}$, $z \in [1, N]$, which denotes the i-th feature map of the k-th layer of the z-th picture. The size r_k of the feature map obtained between different layers is different. The inputting size of picture of each layer is $C_k - R_k + 1$.

(4) Combine the feature maps of the same location corresponding to different pictures into a set $P_{k,i}$, which indicates the i-th feature map of the k-th layer, and sample the $R_k \times R_k$ window size of all the graphs in the set to obtain the image set $P_{k,i}$. And the number of sampling result in $P_{k,i}$ is $n \times (r_k - R_k + 1) \times (r_k - R_k + 1)$.

(5) Calculating all the images in the set $P_{k,j}$ according to the 2DPCA steps in 3.3.1, then obtain the corresponding projection matrix $U_{k,j}$, $V_{k,j} \in R^{R_k \times R_k}$, which correspond to the two columns and rows, and the feature value sets λ_u and λ_v corresponding to each feature vector.

(6) After arbitrarily adding the eigenvalues in λ_u and λ_v get a new set λ^*, take the largest D_k values before. Every value corresponds to the two eigenvectors ξ_u^a and ξ_v^b respectively. Calculating $\xi_v^b * \xi_u^{aT}$ in turn yields a set of evaluation matrices $M = \{M_1, M_2, \cdots M_T\}$, where the maximum value of T is R_k^2. Then the initialize the convolution kernel in turn.

3.3 Parametric Equalization Normalization

For an N-layer convolutional neural network, the loss function is defined as $\ell(z_N)$. We usually want the gradient of the weight $W_k\,(i,j)$ of the k-th layer to satisfy the following form:

$$C_{k,i,j}^2 = E_{z_0} \sim D\left[\frac{\partial}{\partial W_k\,(i,j)}\ell\,(z_N)\right] = E_{z_0} \sim D\left[z_{k-1}\,(j)\,\frac{\partial}{\partial z_k\,(i)}\ell\,(z_N)\right] \quad (5)$$

where D is the set of input images and $y_k = \frac{\partial}{\partial z_k}\ell\,(z_N)$ is the backpropagation error. A similar formula can also be applied to offset b_k, except that the coefficient becomes constant 1.

In order to make all parameters be able to learn at the same speed, we need to scale (5) proportionally, hoping to be a constant for all weights:

$$\overline{C}_{k,i,j}^{2} = \frac{C_{k,i,j}^{2}}{\|W_k\|_2^2} \tag{6}$$

Among them, $\|W_k\|_2^2$ represents the 2-norm square of matrix W_k, but due to the effect of nonlinear activation function, this condition is difficult to control and guarantee, and the change of weight will directly affect the final output value y_k. We therefore need to simplify (6) so that each convolution kernel in the same layer W_k satisfies about a constant, rather than strictly for all weights:

$$\overline{C}_{k,j}^{2} = \frac{1}{N}\sum_{i}\overline{C}_{k,i,j}^{2} = \frac{1}{N\|W_k\|_2^2}E_{z_0} \sim D\left[z_{k-1}^{2}(j)\,\|y_k\|_2^2\right] \tag{7}$$

where N is the number of rows of the weight matrix. This formula makes all the values in the same weight matrix have the same trend. At the same time, we can note that for the input in the layer has little effect on the gradient in the entire network. It can be seen that $z_{k-1}(j)$ and $\|y_k\|$ are independent of each other, so we can further simplify the objective function:

$$\overline{C}_{k,j}^{2} \approx E_{z_0} \sim D\left[z_{k-1}(j)^2\right]\frac{E_{z_0} \sim D\left[\|y_k\|_2^2\right]}{N\|W_k\|_2^2} \tag{8}$$

The method taking the approximate value is convenient to adjust the change rate of the weight of each layer.

Intra-layer Equalization Normalization. For the sake of clarity, the pseudo-code of the intra-layer equalization normalization is shown in Algorithm 1. For all hidden layers $k \in \{1, 2, \cdots, N\}$ in the network, we calculate the mean and standard deviation of all output values, and make all the output values satisfy the unit mean β and unit variance, that is, calculate the average value $\hat{\mu}_k(i)$ and the variance value $\hat{\sigma}_k(i)^2$ of the output value of each channel $z_k(i)$ first, and then the weights W_k and offsets b_k are divided by the coefficients respectively to make adjustment.

$$W_k(i,:) \leftarrow W_k(i,:)/\hat{\sigma}_k(i) \tag{9}$$

$$b_k(i) \leftarrow \beta = \hat{\mu}_k(i)/\hat{\sigma}_k(i) \tag{10}$$

Inter-layer Equalization Normalization. The parameter adjustment method in Intra-layer Equalization Normalization makes the output of each layer satisfy a variance of 1, and for all the rate of change $C_{k,i}^2$ in W_k is a constant. However, it does not guarantee the rate of change between layers. Here we use an iterative method to make the rate of change $C_{k,i}^2$ between all layers be a constant. The pseudo-code of crossover operator is shown in Algorithm 2.

Algorithm 1 Intra-layer Equalization Normalization

```
Input: the number of layers $N$
Output: the weights $W_k$ and offsets $b_k$
Randomly initializes weights $W_k$
Set all the offsets as $b_k$=0
Select a part of the sample data $z_0 \in \overline{D} \subset D$ from the training data set;
for all $z_0 \in \overline{D}$ do
    Obtain the output of each layer, calculate the mean $\hat{\mu}_k(i)$ and variance $\hat{\sigma}_k^2\,(i)$ of each channel of the output;
    Update the scale of the weight $W_k\,(i,:)$ based on Equation (9)
    Update the value of offset $b_k\,(i)$ based on Equation (10)
end for
return $W_k$, $b_k$
```

Algorithm 2. Inter-layer Equalization Normalization

```
Input: the number of layers $N$
Output: the weights $W_k$ and offsets $b_k$
Randomly initializes weights $W_k$
Set all the offsets as $b_k$=0
Select a part of the sample data $z_0 \in \overline{D} \subset D$ from the training data set;
for all $N$ layers do
    Calculate the ratio $\overline{C}_k = E_j\left[\overline{C}_{k,j}\right]$ for each layer;
    Calculate a scale change coefficient $r_k = \left(\bar{C}/\bar{C}_k\right)^{\alpha/2}$, $\alpha$ is an attenuation factor;
    Update the weight and offset of each layer: $W_k \leftarrow r_k W_k$ and $b_k \leftarrow r_k b_k$£
end for
return $W_k$, $b_k$
```

3.4 Convolution Kernel Initialization Procedure Based on 2DPCA and Equalization Normalization

Assume that there are a total of n categories of picture data to be trained, the size of the convolution kernel of each convolution layer k is $R_k \times R_k$, the number of convolution kernels per layer is p_i, and N is the number of convolutional layers which determined by the model:

(1) Get the evaluation matrix set M according to the 2DPCA initialization process in Sect. 3.2;

(2) If the number of sets M is greater than or equal to the number of convolution kernels p_k, initialize all the p_k convolution kernels by taking the matrix corresponding to the previous p_k eigenvalue according to the value size λ^*; If the number of feature vectors d_k is less than the number of convolution kernels p_k, initialize the first d_k convolution kernels with all the assignment matrices, and the remaining uninitialized convolution kernel roulette algorithm randomly selects the assignment matrix into the weight matrix W_k of the current convolution layer;

(3) Using Algorithm 1, each convolutional layer is calculated separately, and calculate the mean $\hat{\mu}_k$ and variance $\hat{\sigma}_k^2$ of p_k convolution kernels within each layer;

(4) Using Algorithm 2, the entire set of weight matrixes are firstly extracted in the entire network model. The dimensions of each weight are different. The number of columns and rows in the matrix are the size and the number of the k-th layer convolution kernel, respectively. Then the iterative operation of the weight adjustment is performed:

(4.1) Traverse all N weight matrices and calculate $\overline{C}_k$ for each layer;
(4.2) Calculate the global average ratio $\bar{C}$;
(4.3) Then calculate the scale factor r_k;
(4.4) Adjust the weight W_k and bias b_k.

The above operations are iterated until the weight adjustment approaches convergence, and a new set of weight matrices $\{W'_1, W'_2, \cdots, W'_N\}$ is obtained and assigned (approximately 10 iterations).

4 Experiments and Results

The experiment of this article is trained on Alibaba Cloud Server. The CPU model is Intel Xeon Platinum 8163 (dual core). The processor is clocked at 2.5 GHz. It uses only the CPU for training. The operating system is Windows server 2012R, and the memory size is 8 GB. The programming language is c/c++.

4.1 Histogram Analysis

The output values of six hidden layers are counted in turn. The activation function uses the *Tanh* and *Relu* functions to draw the distribution histograms of the output values. Their histograms respectively as shown in Fig. 1 and Fig. 2, respectively.

From Fig. 1 and Fig. 2, it is relatively reasonable for the distribution of each layer of the two models. The distribution of the output values of each layer is not much different, and the parameters of back propagation adjustment are all better, which makes the model easier to training. At the same time, since the algorithm is based on the specific training sample data after the principal component analysis is performed for initialization, the initial value of the convolution kernel is more suitable for this sample data, and there will be a good starting point for the optimization of such nonconvex functions.

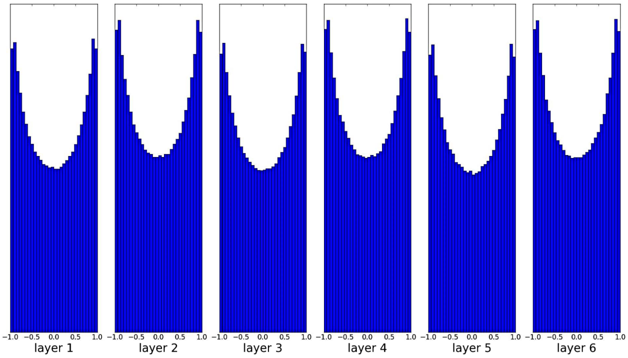


Fig. 1. The output distribution histogram with *Tanh* activation function.

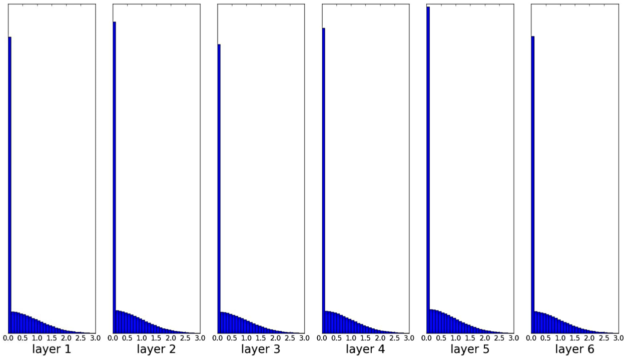


Fig. 2. The output distribution histogram with *Relu* activation function.

4.2 Model Comparison

In order to verify the performance of the initialization algorithm proposed in this chapter, the algorithm was applied to the classical hand-written digital data set MNIST for verification. *Relu* and *Tanh* were used as activation functions to calculate the accuracy rate after each training. The experimental results were compared with other classical Gaussian random initialization, Xavier initialization, and MSRA initialization methods to detect the rate of increase in recognition rate. Draw a graph of the accuracy rate of the training process.

The experiment adopts the classical Lenet-5 convolutional neural network model. The model consists of two convolutional layers and two alternating pooling layers, finally connects to a fully connected layer. Among them, the convolution kernel size is 5×5, the step length is 1, the pooling window size is 2×2, and the step size is also 2. That is, using the non-overlapping pooling and the pooling method is the maximum pooling. Whats more, the dropout regularization mechanism is added in the full connection layer to enhance the generalization ability of the network. The batch size is 1, that is, every time an image is passed in, it will adjust weight values by back-propagating and calculate the error. In order to make comparison images more clearly and avoid the interference caused by the curve crossover, calculate the overall accuracy rate after every 10 training charts, that is, the accuracy rate equels the correct number of identification pictures/the number of trained pictures. The total number of iterations is 400, that is, the curve obtained by training 4000 images. Each algorithm counts 5 times and takes the average value.

When the activation function takes *Relu*, the comparison curve of the accuracy during the training of the three different initialization methods is shown in Fig. 3.

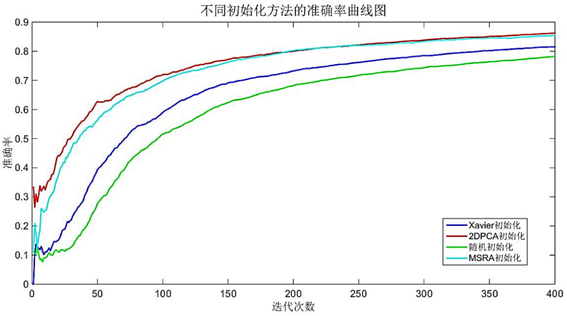


Fig. 3. The accuracy curves of different initialization methods with *Relu* activation function.

This experiment mainly contrasts the rising rate of recognition rate during training, that is, the convergence speed of the network model. From this experimental comparison chart, it can be seen that the accuracy of the four initialization methods rises as the training progresses. The recognition rate of the 2DPCA initialization method proposed in this paper rises fastest in the early stage, and it can reach the accuracy rate of 0.3 at the beginning of the training. However, the accuracy of the other methods is only 0.1 at the beginning. And for the 10 classification problem, the probability of random selection is 0.1. It can be seen that since the 2DPCA initialization is based on the actual data image initialization, a weight value suitable for the sample data can be initialized directly, and the optimization is started from a good starting point, and the accuracy rate is increased faster. And the other three methods are not based on sample data. The MSRA initialization effect is relatively good, it proves that it is indeed suitable for *Relu* activation function (Fig. 4).

It can be seen from this experiment that MSRA and Xavier have similar effects, when Tanh is used as an activation function. Since MSRA is not particularly suitable for *Tanh* functions, the effect is lower than *Relu* as an activation function, and the Xavier effect on the Tanh function is slightly higher than the MSRA initialization. Among them, the effect of 2DPCA initialized based on the sample data is still the best, and the accuracy rate of the initial period is the

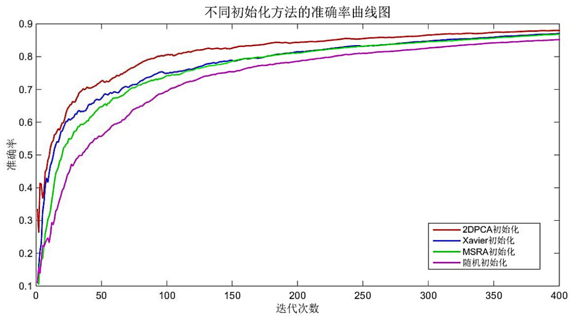


Fig. 4. The accuracy curves of different initialization methods with *Tanh* activation function.

fastest. It can be seen that the 2DPCA initialization method both has a good effect in the *Relu* activation function and the Tanh activation function, and is not limited by the type of the specific activation function.

5 Conclusion

This paper studies the initialization method of convolutional neural network, statistics the output value of each layer under different initialization methods and draws a histogram, and analyzes the distribution of output values from the histogram. 2DPCA-based convolution kernel initialization method is proposed for the problem of its distribution. 2DPCA is used to extract key features from the sample data and initialize the convolution kernel, and an equalization normalization method is introduced to adjust the size of the weights between the layers. The method does not need to manually set hyper-parameters, avoids random values, and does not have limitations on the types of activation functions. The parameters are completely determined according to the characteristics of specific training sample data. Finally, the histogram distribution and the curve comparison diagram of the model training show that the proposed method can effectively avoid the uncertainty caused by initializing the weights and accelerate the training speed of the entire model.

Acknowledgements. This work is supported by the National Key Research and Development Project (No. 2016YFC0401607), the Fundamental Research Funds for the Central Universities (No. 2019B22314), and the National Key R&D Program of China (No. 2018YFC0407101).

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034 (2015)

2. 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)
3. Liu, X., Wang, J., Xu, K.: Novel image feature extraction algorithm based on fusion autoencoder and CNN. Appl. Res. Comput. **34**(12), 3839–3843 (2017)
4. Pacheco, A.G., Krohling, R.A., da Silva, C.A.: Restricted boltzmann machine to determine the input weights for extreme learning machines. Expert Syst. Appl. **96**, 77–85 (2018)
5. Seyfioğlu, M.S., Gürbüz, S.Z.: Deep neural network initialization methods for micro-doppler classification with low training sample support. IEEE Geosci. Remote Sens. Lett. **14**(12), 2462–2466 (2017)
6. Sun, W., Su, F., Wang, L.: Improving deep neural networks with multi-layer maxout networks and a novel initialization method. Neurocomputing **278**, 34–40 (2018)
7. Tang, J., Wang, D., Zhang, Z., He, L., Xin, J., Xu, Y.: Weed identification based on k-means feature learning combined with convolutional neural network. Comput. Electron. Agric. **135**, 63–70 (2017)
8. Thimm, G., Fiesler, E.: Neural network initialization. In: Mira, J., Sandoval, F. (eds.) IWANN 1995. LNCS, vol. 930, pp. 535–542. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-59497-3_220
9. Yang, N., Li, Y., Yang, Y., Zhu, M.: Convolutional neural networks based on sparse coding for human postures recognition. In: AOPC 2017: Optical Sensing and Imaging Technology and Applications, vol. 10462, p. 104622B. International Society for Optics and Photonics (2017)
10. Yunfei, L., Randi, F., Wei, J., Nian, J.: Image super-resolution using multi-channel convolution. J. Image Graphics **22**(12), 1690–1700 (2017)
11. Zhang, Z., Ji, J.: Classification method of FMRI data based on convolutional neural network. Pattern Recog. Artif. Intell. **30**(6), 549–558 (2017)
12. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. pp. 249–256 (2010)

Research on $PM_{2.5}$ Integrated Prediction Model Based on Lasso-RF-GAM

Tingxian Wu[1], Ziru Zhao[1], Haoxiang Wei[2], and Yan Peng[1](✉)

[1] School of Management, Capital Normal University, Beijing, China
pengyan@cnu.edu.cn
[2] School of Engineering, University of Washington Seattle, Seattle, USA

Abstract. $PM_{2.5}$ concentration is very difficult to predict, for it is the result of complex interactions among various factors. This paper combines the random forest-recursive feature elimination algorithm and lasso regression for joint feature selection, puts forward a $PM_{2.5}$ concentration prediction model based on GAM. Firstly, the original data is standardized in the data input layer. Secondly, features were selected with RF-RFE and lasso regression algorithm in the feature selection layer. Meanwhile, weighted average method fused the two feature subsets to obtain the final subset, RF-lasso-T. Finally, the generalized additive models (GAM) is used to predict $PM_{2.5}$ concentration on the RF-lasso-T. Simulated experiments show that feature selection allows GAM model to run more efficiently. The deviance explained by the model reaches 91.5%, which is higher than only using a subset of RF-RFE. This model also reveals the influence of various factors on $PM_{2.5}$, which provides the decision-making basis for haze control.

Keywords: $PM_{2.5}$ concentration · Random Forest-RFE · Lasso method · Feature fusion · GAM model

1 Introduction

In recent years, PM $_{2.5}$ concentration has increased astronomically on almost every continent, and studies show that the damage done are catastrophic and some are even irreversible. Not long-ago the data collected by the Chinese Ministry of Environmental protection presented a sharp increase in both $PM_{2.5}$ concentration and cardiovascular disease. Researchers have shown that high $PM_{2.5}$ concentration is the main cause of lung cancer, respiratory disease, and metabolic disease [1].

Predicting $PM_{2.5}$ concentration has been done in article [2], Professor Joharestani and Professor Cao collected $PM_{2.5}$ air pollution data and climatic features. With Random Forest Modeling and Extreme Gradient Boosting, they were able to form a model to predict $PM_{2.5}$ in the certain areas. But unfortunately, meteorological phenomes caused their data inaccuracy. The data in our experiment was professionally measured and provided by China's National Population and Health Science Data Sharing Service Platform.

The rest of the paper is arranged as the following: In the second part of this paper, the relevant research work is introduced, and in the third part, a $PM_{2.5}$ concentration

Y. Tan et al. (Eds.): DMBD 2020, CCIS 1234, pp. 83–94, 2020.
https://doi.org/10.1007/978-981-15-7205-0_8

prediction model is proposed. This model combines RF-RFE method and lasso model for joint feature selection. Then, the $PM_{2.5}$ concentration prediction model based on GAM is constructed. The model is verified by using the meteorological monitoring data and pollution index monitoring data of the Ministry of Environmental Protection. The experimental results show the effectiveness of the integrated model. In the fourth part, it is summarized that the model can be used to predict the concentration of $PM_{2.5}$, which is helpful for the relevant departments to better understand the influencing factors of $PM_{2.5}$ concentration and provide auxiliary decision-making basis for air pollution control.

2 Related Research Work

2.1 RF-RFE Method

RF(Random forest) and RF-RFE(Random Forest-Recursive Feature Elimination algorithm) are both descendants of machine-learning. RF is compatible with high-dimensional problems and can predict nonlinear relationships with the downside of its frequent inability to identify strong predictors in the presence of other correlated predictors [3].

Random forest can be used for parallel operation, regression analysis, classification, unsupervised learning, and other statistical data analysis; even when there is a large proportion of data missing from the data set, it has the ability to estimate the absent data value, and keep the accuracy unchanged. Based on the above characteristics, this study will select random forest for RFE and RF-RFE for feature selection of PM2.5 concentration influencing factors.

2.2 Lasso Regression

The lasso method (Least Absolute Shrinkage and Selection Operator) was first introduced by Professor Robert Tibshirani. The Lasso method which include the regression shrinkage and selection. It is said to "minimize the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant." [4] Ultimately, Lasso has the ability to achieve two main tasks, regularization and feature selection. In a recent study with Professor Lingzhen Dai and his colleagues, they used the adaptive Lasso method to identify PM2.5 components associated with Blood Pressure in Elderly man. [5] Adaptive Lasso is a more recent version of Lasso, it uses weight to penalize different predictors and allows the researchers in this study to identify the right subset model and satisfy asymptotic normality.

For the given data $\left(X^i, y_i\right), i = 1, 2, \ldots N, X^i = \left(X_{i1}, \ldots X_{ip}\right)^T$ are the predictor variables and y_i are the responses. The mathematical expression of lasso method is

$$\left(\hat{\alpha}, \hat{\beta}\right) = arg\ min\left\{\sum_{i=1}^{N}(y_i - \alpha - \sum_j \beta_j x_{ij})^2\right\}, subject\ to \sum_j |\beta_j| \le t. \tag{1}$$

Where, t ≥ 0 is a tuning parameter. α is the penalty parameter, which has a negative correlation with the number of features finally selected; β is the coefficient corresponding to the characteristic words in each column of the independent variable x. L1 regularization adds the L1 norm of coefficient β as the penalty term to the loss function, because the regular term is non-zero, thus forcing the coefficient corresponding to the weak feature to become 0.

2.3 Generalized Additive Models (GAM)

In order to reach a better understanding of GAM (Generalized Additive Models), one should be familiar with its close relative GLM (Generalized Linear Models). GLM is a generalization for logistic regression in a linear regression.

The general form of GAM is:

$$\log(\lambda) = \alpha + \sum_{i=1}^{P} f_i(x_i) \tag{2}$$

Where f_i represents smoothing functions such as smooth spline, natural cubic spline and local regression [6].

3 $PM_{2.5}$ Integrated Prediction Model Based on Feature Selection and GAM Model

The RF-RFE-lasso and GAM algorithms are used to form the $PM_{2.5}$ integrated prediction model.

The model structure is shown as Fig. 1.

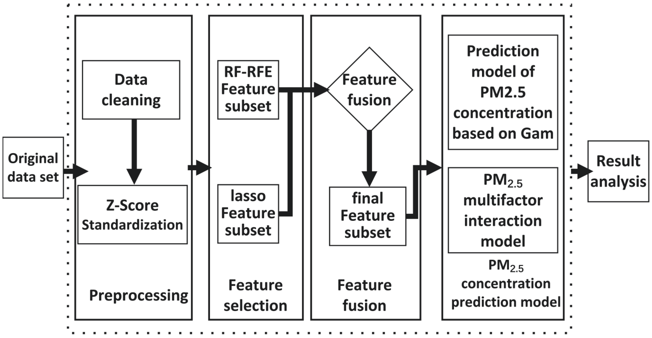


Fig. 1. $PM_{2.5}$ concentration integration model

3.1 Data Preprocessing

3.1.1 Data Source

The experimental data are provided by China's National Population and Health Science Data Sharing Service Platform and released by the Ministry of Environmental Protection. (http://www.ncmi.cn/). Data Set 1 is the air pollution index of provincial capitals from January 1, 2015 to December 31, 2016. Data set 2 is the surface meteorological data from January 1, 2015 to December 31, 2016. The variables selected for this specific experiment are listed in Table 1.

Table 1. Parameter information

Variable	Parameter	Unit	Period
X1	PM_{10}	$\mu g/m^3$	2015.1–2016.12
X2	SO_2	$\mu g/m^3$	
X3	NO_2	$\mu g/m^3$	
X4	O_3	$\mu g/m^3$	
X5	Pressure	0.1 hPa	
X6	Pressure max	0.1 hPa	
X7	Pressure min	0.1 hPa	
X8	Temperature	0.1 °C	
X9	Temperature max	0.1 °C	
X10	Temperature min	0.1 °C	
X11	Relative humidity	1%	
X12	Relative humidity min	1%	
X13	Daily rainfall	0.1 mm	
X14	Wind speed	0.1 m/s	
X15	Wind speed max	0.1 m/s	
X16	Direction of Wind speed max	–	
X17	Wind speed extreme max	0.1 m/s	
X18	Direction of Wind speed max	–	
X19	sun	0.1 h	

3.1.2 Data Standardization

As presented in Table 1, each data item has different dimensions. When joint analysis is carried out, Z-Score standardization method is selected for dimensionless standardization processing, as shown in Formula (3).

$$Znorm(x_i) = \frac{x_i - \overline{X}}{\sigma(X)}. \tag{3}$$

Where $\boldsymbol{x_i}$ represents the original value, $\overline{\boldsymbol{X}}$ represents the mean value of the original data, $\boldsymbol{\sigma}(\boldsymbol{X})$ is the standard deviation of the original data, and $\boldsymbol{Znorm}(\boldsymbol{x_i})$ is the standardized result.

3.2 Feature Selection

The complex nonlinear relationship between various indexes of air quality monitoring and $PM_{2.5}$ concentration and the multicollinearity among air quality indexes affects the performance of the model. For our specific experiment, the RF-RFE algorithm is utilized alongside Lasso algorithm to select a feature subset with stronger correlation with the results, so as to improve the prediction accuracy and efficiency of the model.

3.2.1 Feature Selection Based on RF-RFE

The RF-RFE algorithm is used to filter the 19 variables in Table 1, and the resulting feature subset is the subset corresponding to the minimum root mean square error (RMSE).

The implementation process of the RFE method is as follows:

(1) Construct a feature matrix with the given feature vectors in the data set, each row represents a sample and each column corresponds to a feature;
(2) Set the control parameters for constructing RFE function and adopt random forest function as well as the 20-fold cross-validation sampling method;
(3) RFE algorithm is used to sort these features according to their correlation with $PM_{2.5}$ concentration;
(4) Based on the ranking results, the first N (N is the number of features meeting the user's needs) features with the highest correlation with $PM_{2.5}$ concentration are selected as the featured subset.

The final number of features is 9, namely PM_{10}, NO_2, RH_ave, SO_2, RH_min, wind_ex, sun, O_3, and wind_max. As shown in Fig. 2.

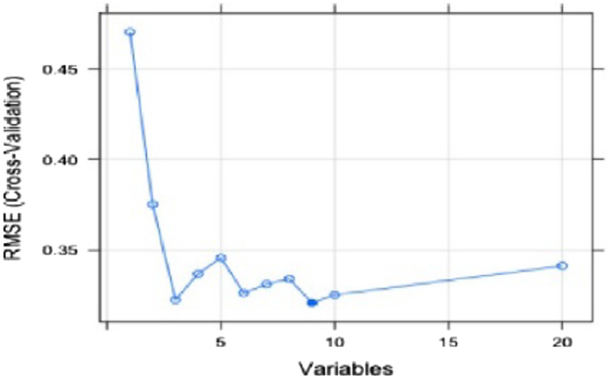


Fig. 2. RF-RFE result graph

3.2.2 Feature Selection Based on Lasso

Lasso estimation is carried out on all independent variables, and the change process of regression coefficient is shown in Fig. 3.

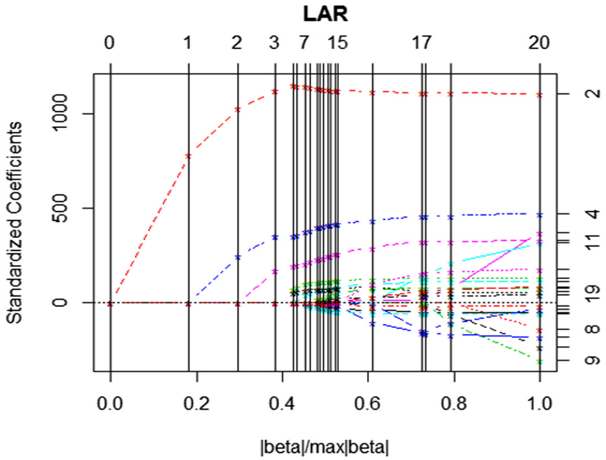


Fig. 3. Lasso estimation path map

In the figure, the abscissa serves as the value of β, or the number of steps, and the parameters represented by the ordinate appear as the vertical lines. The vertical line in the figure corresponds to the number of iterations in lasso.

The final selected feature subset is shown in Table 2.

Table 2. Feature selected with Lasso

Variable	PM_{10}	NO_2	RH_ave	RH_min	SO_2
Range	1.00	2.00	3.00	4.00	5.00
Variable	O3	wind_direction_max	season	wind_ave	wind_direction_ex
Range	6.00	7.00	8.00	9.00	10.00
Variable	precipitation	wind_max	pre_ave	wind_ex	temp_min
Range	11.00	12.00	13.00	14.00	15.00

3.2.3 Feature Subset Fusion

Common feature subset fusion methods include feature element intersection method and weighted fusion method. The general form of the weighted fusion formula is.

$$\frac{(a_1 * mathod1 + a_2 \ldots a_n * methodn)}{n} \tag{4}$$

a_1, a_2, etc. are the weighting coefficients of the feature subset, where $a_1 + a_2 + \ldots a_n = 1$.

The weighted average method is used to obtain the fused subset of the RF-RFE and the lasso feature subsets. The process is as follows: in line with the ordering of each feature in the RF-RFE feature subset and the lasso subset, the corresponding score is assigned in ascending order based on rank. The weights coefficients of both subsets are set to 50% and the total score of each feature is added and averaged. Finally, the top 12 features with the highest average score are selected as the final feature subsets and given the corresponding name. RF-lasso-T, as shown in Table 3:

Table 3. Joint feature subset

Serial number	Features	Score	Average value
1	PM_{10}	15	7.5
2	NO_2	14	7
3	RH_ave	26	13
4	SO_2	23	11.5
5	RH_min	23	11.5
6	O_3	18	9
7	wind_ex	12	6
8	wind_max	11	5.5
9	sun	9	4.5
10	wind_direction_max	9	4.5
11	season	8	4
12	wind_ave	7	3.5

3.3 Construction of $PM_{2.5}$ Concentration Prediction Model

3.3.1 Prediction Model of $PM_{2.5}$ Concentration and Influencing Factors

Based on the joint feature subsets obtained by RF-RFE and lasso methods, PM_{10}, SO_2, NO_2 concentration and other explanatory variables were introduced into the model. The effects of explanatory variables are eliminated as a result of a smoothing spline function while seasonal dummy variables are introduced to eliminate periodic effects. The degree of freedom is selected by determining the smallest sum of the absolute values of the model partial autocorrelation (PACF). The GAM model constructed by the joint feature subset RF-lasso-T is shown in formula (5):

$$Y(PM2.5) = \sum_{i=1}^{j} s(X) + \alpha. \tag{5}$$

Where Y is the concentration of $PM_{2.5}$, X refers to the variable, I is the serial number of each variable, J is the number of variables, and S is the smooth function of the model. The entire model adopts the Gaussian iteration method.

3.3.2 Analysis of Prediction Results

In agreement with the 80% and 20% proportion, the whole data set is divided into a training set and a test set.

On the training set, two GAM models were constructed, one being with the feature fusion subset RF-lasso-T and the other using the full feature subset. The execution duration of the two models is compared shown as in the following figure (Fig. 4):

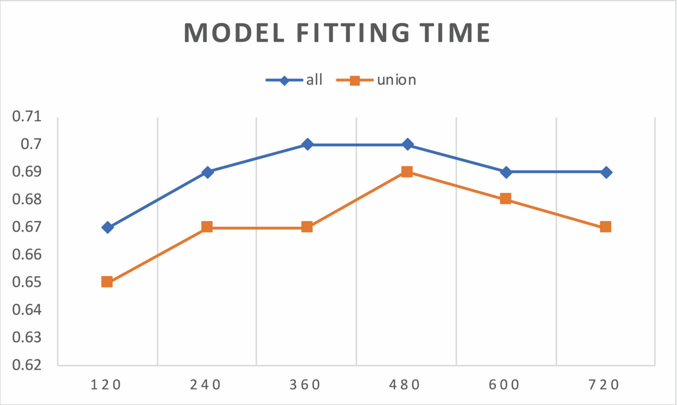


Fig. 4. Model time-consuming comparison chart

Evidently, the prediction time of the integrated GAM model is lower than that of the GAM model only whereas the deviance explained by GAM model based on RF-lasso-T is 91.5%, which is higher than that of GAM model based on RF-RFE (90.9%). The test results of RF-lasso-T-GAM model are shown in Table 4:

Table 4. RF-lasso-T GAM model hypothesis test results

Smooth factor	Coefficient	F	P
s(PM10)	**7.932**	**77.965**	**<2.00E−16**
s(NO2)	**6.81**	**13.437**	**<2.00E−16**
s(RH_ave)	**5.745**	**6.716**	**0.000000154**
s(SO2)	**1.768**	**6.836**	**0.000861**
s(RH_min)	6.637	2.26	0.019107
s(O3)	**1.524**	**23.223**	**7.36E−10**
s(wind_ex)	**8.395**	**2.62**	**0.004357**
s(wind_max)	2.816	1.226	0.420072
s(sun)	1	0.161	0.688202
s(wind_direction_max)	1	1.297	0.255324
s(wind_ave)	6.39	1.776	0.072678

Note: the bold ones are significant influence factors.

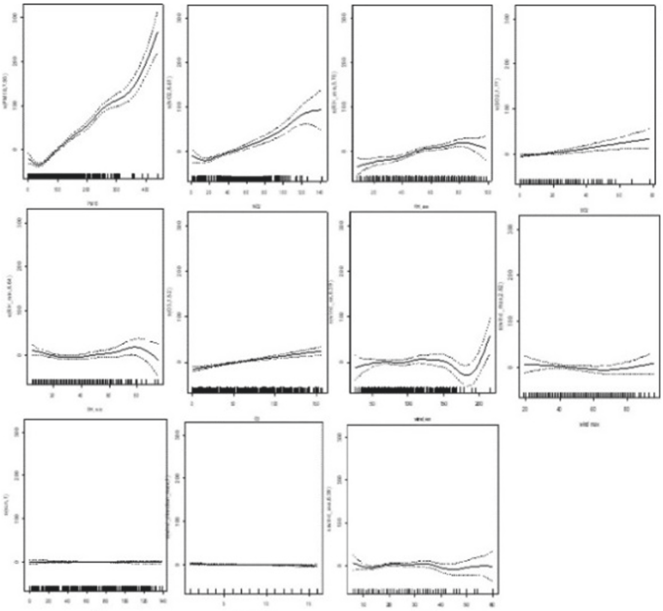


Fig. 5. Effect chart of $PM_{2.5}$ influencing factors

Figure 5 displays the effect of the influencing factors on $PM_{2.5}$. Dotted line in the figure shows the point by point standard deviation of fitting variables, i.e. the upper and the lower limits of signal interval; solid line portrays the smooth fitting curve of $PM_{2.5}$ concentration. Abscissa represents the measured value of each influence factor while ordinate denote the smooth fitting value of influence factors on $PM_{2.5}$ concentration. The value in ordinate brackets represents the estimated freedom value. The results show that the concentrations of PM_{10}, NO_2, RH_ave, RH_min, wind_ex, wind_max, wind_ave and $PM_{2.5}$ boast a non-linear relationship, while the concentrations of SO_2, O_3, sun, wind_direction_max and $PM_{2.5}$ consists of a linear relationship.

The overall data set was divided into training and test sets based on the ratio of 80% and 20%. The training model was used to simulate a 146-day data in the test set, and the $PM_{2.5}$ daily concentration value was predicted. The model prediction fit is shown in Fig. 6. The average value of $PM_{2.5}$ is 76.25, the average value of the predicted $PM_{2.5}$ is 76.27, and the root mean square error is 0.377.

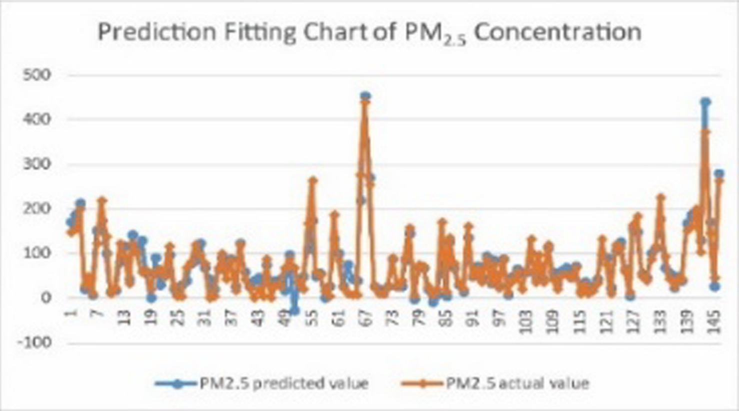


Fig. 6. Prediction fitting chart of $PM_{2.5}$ concentration

3.3.3 Analysis of Interaction Model of Influencing Factors

The change in $PM_{2.5}$ concentration can be affected by the interactions between influencing factors. In pursuance of the relationship between interactions and the concentration of $PM_{2.5}$, the smooth spline function is used to connect the influencing factors. With the combination of various pollutants and meteorological factors, an RF-lasso-T-GAM model was established. The deviance explained by the interaction model was 95.7% with an adjustment decision coefficient of 0.947. The results of the model show that cross variables such as PM_{10} and NO_2, PM_{10} and RH_ave, PM_{10} and O_3, PM_{10} and sun, NO_2 and SO_2, NO_2 and RH_min, NO_2 and O_3, NO_2 and wind_ex, RH_ave and SO_2, RH_min and O_3, RH_ave and O_3, RH_ave and wind_ex are significant at the level of $P < 0.001$. The cross terms include the interaction between air pollutants SO_2, NO_2, PM_{10} and meteorological elements as well as the interaction among air pollutants. All the above observations indicate that the change in $PM_{2.5}$ concentration is affected by the interaction between air pollutants and meteorological elements.

Taking the remarkable interaction between NO_2 and various factors as an example, the interaction model is visually plotted and shown in Fig. 7.

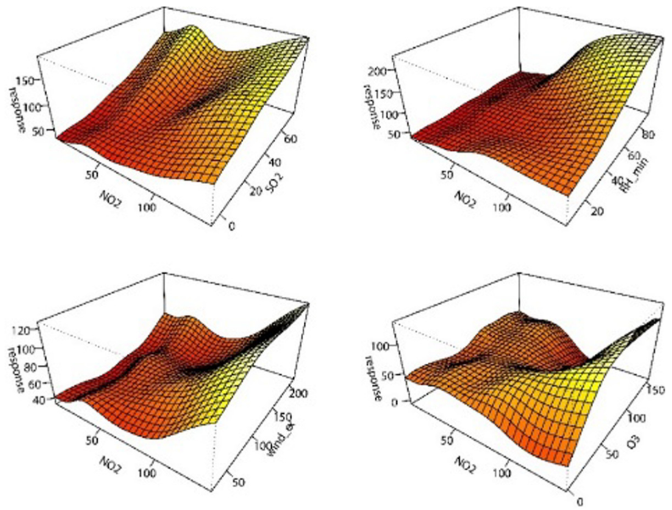


Fig. 7. Effect of interaction of influencing factors on $PM_{2.5}$ concentration

As can be seen from Fig. 7(1), when the NO_2 concentration is constant, with the increase of SO_2, the $PM_{2.5}$ concentration increases first, then decreases and then increases, which demonstrates a wave-like upward trend. When SO_2 is constant, with the increase of NO_2 concentration, the concentration of $PM_{2.5}$ shows a trend of increasing first, then decreasing and then increasing. From Fig. 7(2), when NO_2 concentration is constant, with the increase of RH_min, the concentration of $PM_{2.5}$ increases first, then decreases and then increases, but the overall trend is relatively stable. When RH_min is constant, with the increase of NO_2 concentration, the concentration of $PM_{2.5}$ shows a wave-like upward trend of increasing first, then decreasing and then increasing. From Fig. 7(3), when NO_2 concentration is constant, with the increase of wind_ex, the concentration of $PM_{2.5}$ decreases first and then increases, then increases again after a large decrease. When wind_ex is constant, with the increase of NO_2 concentration, the concentration of $PM_{2.5}$ decreases first and then increases, then increases again after a large decrease. As can be seen from Fig. 7(4), when the NO_2 concentration is constant, with the increase of O_3, the $PM_{2.5}$ concentration increases sharply first and then decreases, and then gradually increases in a wave shape. When O_3 is constant, with the increase of NO_2 concentration, the concentration of $PM_{2.5}$ shows a wave trend of decreasing first and then increasing, then there is a large decrease, and then it continues to rise.

4 Conclusion

In this study, RF-RFE and Lasso method are used to select the characteristics of the air quality data set. The runtime efficiency of the RF-lasso-T based on GAM model is higher than the GAM model constructed directly on data sets without feature selection and the deviance explained by this model is higher than the GAM model using a single feature subset of RF-RFE. The fitting results of this model can be further analyzed to obtain the meteorological and pollution factors with significant influence on $PM_{2.5}$, as well as the linear and nonlinear relationship between meteorological factors and $PM_{2.5}$ concentration. Visual results are provided to support auxiliary decision-making basis for $PM_{2.5}$ prediction and air pollution control.

Funding. This work was supported by the National Education Sciences Planning-Key Program of Ministry of Education (DLA190426).

References

1. Amsalu, E., Wang, T., Li, H., et al.: Acute effects of fine particulate matter ($PM_{2.5}$) on hospital admissions for cardiovascular disease in Beijing, China: a time-series study. Environ. Health **18**(70), 1–12 (2019)
2. Joharestani, M.Z., et al.: $PM_{2.5}$ prediction based on random forest, xGBoost, and deep learning using multisource remote sensing data. Atmosphere **10**(7), 364–373 (2019)
3. Darst, B.F., Malecki, K.C., Engelman, C.D.: Using recursive feature elimination in random forest to account for correlated variables in high dimensional data. BMC Genet. **19**(65), 1–6 (2018)
4. Tibshirani, Robert: Regression shrinkage and selection via the lasso. J. Roy. Stat. Soc.: Ser. B (Methodol.) **58**(1), 267–288 (1996)
5. Dai, L., Mehta, A., Mordukhovich, I., Just, A., et al.: Differential DNA methylation and PM $_{2.5}$ species in a 450 K epigenome-wide association study. Epigenetics **12**(2), 139–148 (2016)
6. Hastie, T., Tibshirani, R.: Generalized additive models. Stat. Sci. **1**(3), 297–310 (1986)

Research on Short-Term Urban Traffic Congestion Based on Fuzzy Comprehensive Evaluation and Machine Learning

Yuan Mei[1](✉), Ting Hu[2](✉), and Li Chun Yang[1]

[1] Department of Information and Software Engineering, Chengdu Neusoft University,
Dujianyan, Chengdu 611844, Sichuan, China
2551161628@qq.com, 1377759045@qq.com

[2] Department of Information Management, Chengdu Neusoft University, Dujianyan, Chengdu
611844, Sichuan, China
1187352043@qq.com

Abstract. There are many factors that affect urban traffic flow. In the case of severe traffic congestion, the vehicle speed is very slow, which results in the GPS positioning system's estimation of the vehicle speed being very inaccurate, which in turn leads to poor reliability of the estimated congestion time of the road segment. The main contents of this study are: in the case of urban traffic congestion, the prediction and analysis of the degree of traffic congestion and the length of congestion. Taking the dynamic traffic data of Shenzhen on June 9, 2014 as an example, the road section of Binhe Avenue is selected, and the data of traffic flow, average speed of traffic volume and traffic volume density in the current time period are calculated after data preprocessing, as a measure of traffic. The main impact indicators of congestion status. Then we use the fuzzy comprehensive evaluation method to divide TSI as a traffic congestion evaluation index and divide the road congestion into four levels. In this way, we can get the congestion of the road in each time period of the day and the time required to pass. Then we use the random forest, adaboost, GBDT, Lasso CV and BP neural networks in the machine learning algorithm to build models to measure traffic congestion for training and testing. Finally, the BP neural network has the best effect on this problem, and mean square error is 0.0190. Finally, we used BP neural network to predict and congest the road in the next three hours. From the experimental simulation results, this method can effectively analyze and predict the real-time traffic congestion.

Keywords: Random forest · Adaboost · GBDT · Lasso CV · BP neural network · Fuzzy comprehensive evaluation

1 Introduction

At present, there are many solutions for urban road congestion, such as the use of single and double day limit traffic. But the daily congestion on some roads is still not optimistic. There have been many studies on congestion, such as Fu Gui's short-term traffic flow

Y. Tan et al. (Eds.): DMBD 2020, CCIS 1234, pp. 95–107, 2020.
https://doi.org/10.1007/978-981-15-7205-0_9

prediction model based on support vector machine regression [1], Dewey's short-term traffic prediction based on the combination of K nearest neighbor algorithm and support vector regression [2] but its prediction There is a certain deviation between the result and the true value. Kang Danqing's research on short-term traffic flow prediction methods based on deep learning [3] has better results but is more complicated. To this end, this paper compares the effects of various machine learning algorithms on this problem, and proposes that the back propagation based BP neural network has a better effect on the problem of short-term traffic flow prediction and is simpler than deep learning. The specific research content is as follows.

In this paper, the GPS data was collected in real time on a taxi on the Shenzhen Stock Exchange on June 19, 2014. After preprocessing by deleting invalid data, erroneous data, filling missing data, etc. The traffic flow, average speed of traffic volume and traffic volume density are extracted as the measurement indicators of traffic congestion, and the main influencing factors are determined by calculating the correlation between the indicators. Then through the use of various machine learning algorithms (random forest, adaboost, GBDT, LassoCV, BP neural network), through grid search cross-validation, the best parameters of each algorithm and accuracy under the best parameters are obtained. Comparing the effects of various models, it is obtained that the BP neural network has the highest effect in the current situation, and its mean squared error is 0.0190.

Based on the BP neural network, the BinHe avenue was selected, and a model based on fuzzy comprehensive evaluation to measure the time required for vehicles to pass through the congested road section was established. The congestion situation is divided into four levels: very smooth, unobstructed, congested, and severely congested. The congestion situation on the road section was analyzed on that day. Finally, the TSI evaluation index is used to analyze the effect of the model.

2 Research Methods

2.1 Random Forest Algorithm of CART Tree

CART can be used for regression analysis and classification analysis, and expanded some integrated algorithms based on CART. In order to solve the problem of large data volume and large data volume in a big data environment, this study chose CART as the basic random forest algorithm. The CART decision tree has the advantages of being easy to understand and has some nonlinear classification capabilities, but the single decision tree has some disadvantages. In integrated learning, the above defects can be improved through the random forest integration method. Random forest is composed of many decision trees, and there is no correlation between different decision trees, and the model generalization ability is stronger. The algorithm is mainly to randomly sample the samples, train the decision tree, and then classify the nodes according to the corresponding attributes until they are no longer split, and finally build a large number of decision trees to form a forest [4].

2.2 Adaboost Algorithm

Adaptive Boosting is a boosting method that combines multiple weak classifiers into a strong classifier. Its self-adaptation lies in: the weight of the sample that was divided by the previous weak classifier (the weight corresponding to the sample) will be strengthened, and the sample with the updated weight will be used again to train the next new weak classifier. In each round of training, a new weak classifier is trained with the population (sample population) to generate new sample weights and the speech weight of the weak classifier, and iterates until it reaches a predetermined error rate or the specified maximum number of iterations.

2.3 GBDT Algorithm

Gradient Boosting Decision Tree is a combination of GB (Gradient boosting) and DT (Decision Tree), that is, when a single learner in GB is a decision tree. Decision trees are divided into two categories, regression trees and classification trees. The former is used to predict real values, and the latter is used to classify label values. Through multiple iterations, each iteration generates a weak classifier, and each classifier is trained on the basis of the residuals of the previous classifier. The requirements for weak classifiers are generally simple enough and have low variance and high deviation. Because the training process is to continuously improve the accuracy of the final classifier by reducing the deviation.

2.4 LassoCV Algorithm

Least absolute shrinkage and selection operator is a linear model used to estimate sparse parameters, especially suitable for reducing the number of parameters. For this reason, the Lasso regression model is widely used in compressed sensing. Mathematically, Lasso adds an L1 regular term to the linear model. No matter whether the dependent variable is continuous or discrete, lasso can handle it. In general, lasso has extremely low data requirements, so it is widely used; in addition, lasso can also filter and The complexity of the model is reduced. Variable selection here refers to not putting all variables into the model for fitting, but selectively putting variables into the model to get better performance parameters.

2.5 Neural Network Algorithm

BP neural network is an algorithm that transforms a set of sample input and output problems into nonlinear optimization [5]. It has a three-layer structure and interconnects neurons (Fig. 1).

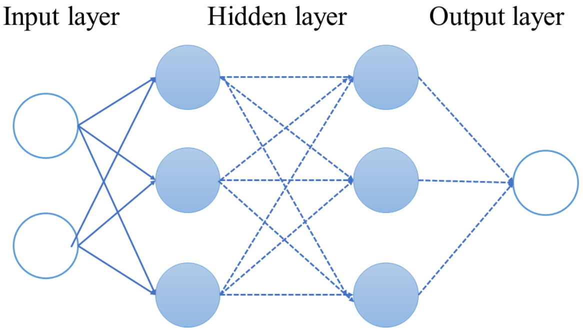


Fig. 1. Neural network structure diagram

The neurons from left to right are the input layer, hidden layer and output layer. The weight of the input layer to the hidden layer (the connection weight of neuron i and neuron j), the threshold; the size of the hidden layer to the output layer may be (the connection weight of neuron i and neuron k).

Usually, two nodes are input, the second layer is hidden, and the output of the three nodes in each layer is one node. The algorithm flow is as follows.

```
Algorithm: BP neural network algorithm

Seeding: Initialize the weight and bias of network.
repeat
    foreach Each training tuple X in D do
        // Forward propagation input
        foreach Each input unit j do
            $O_j = I_j$;// The output of the input unit is its actual input value
        end
        foreach hide or export each cell of the layer j do
            $I_j = \sum_i W_{ij} O_i + \theta_j$; //Regarding the previous layer, the net input of calculation unit j// $O_j = \frac{1}{1+e_j^{-1}}$ ;// Output of calculation unit j
        end
        // Back propagation error
        foreach each unit of the output layer j do
            $Err_j = O_j(1 - O_j) \sum_k Err_k W_{jk}$;//Calculate the error about the next higher layer k
        end
        foreach every right in the network $W_{ij}$
            $\Delta W_{ij} = (l) Err_j O_i$ ;// Weight increment
            $W_{ij} = W_{\overline{ij}} + \Delta W_{ij}$; // Weight update
        end
        foreach $\boldsymbol{\theta_{ij}}$ each bias in the network
            $\Delta\theta_i = (l) Err_j$;//Bias increment
            $\theta_i = \theta_j + \Delta\theta_j$;// Bias update
        end
    end
until termination condition
return forecast result
```

3 Establishing Model

3.1 Selection of Indicators

For the prediction and evaluation of traffic congestion status, we must first select the factor indicators that can accurately and effectively characterize the traffic congestion status. The principle of selection is to have overall completeness, objectivity, operability, and comparability. However, we cannot just pass a certain traffic flow. The parameters evaluate the traffic congestion. The average speed of a vehicle can intuitively represent the state of traffic congestion, but when the vehicle is waiting for a red light at an intersection, although the speed is very small, it does not mean that the road is congested at the moment. Therefore, this paper selects three traffic flow parameters, traffic flow, average speed of traffic flow, and density of traffic flow as factor indicators.

3.2 Calculate Relevance

The average speed of traffic flow refers to the average distance traveled by all vehicles on a road per unit time. This indicator can intuitively reflect the current road traffic congestion. Generally speaking, the greater the speed, the smoother the road; the lower the speed, the more congested the road [6]. Calculated as follows:

$$\overline{\mathrm{v_i}} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{v_I} \tag{1}$$

$\overline{\mathrm{v_i}}$——Average speed of traffic flow (Km/h); N——The number of all vehicles on the road in unit time; $\mathrm{v_i}$——The number of all vehicles on the road in unit time.

Traffic flow density refers to the total number of vehicles on a road per unit length in a unit of time. When the road is congested, the vehicle stalls and the change in traffic flow is almost zero, but the traffic density is very large, so it is decisive for the traffic congestion state. effect. The calculation method is as follows:

$$D = \frac{\mathrm{f}}{\mathrm{v}} \tag{2}$$

D is the required traffic flow density (vehicles/km), and f is the monitored traffic flow every five minutes; v——Average speed.

Calculate the Pearson correlation coefficient between features. The correlation is a non-deterministic relationship, and the correlation coefficient is the amount of linear correlation between the variables studied.

$$r(\mathrm{X}, \mathrm{Y}) = \frac{\mathrm{Cov}(\mathrm{X}, \mathrm{Y})}{\sqrt{\mathrm{Var}[\mathrm{X}]\mathrm{Var}[\mathrm{Y}]}} \tag{3}$$

Among them, Cov (X, Y) is the covariance of X and Y, Var [X] is the variance of X, Var [Y] is the variance of Y.

3.3 Fuzzy Comprehensive Evaluation

Through analysis and selection, and taking traffic flow and average traffic speed as the basic indicators of traffic state characteristics, a weighted calculation model is applied to the traffic state assessment of road segments and road networks, and a road and road network traffic state characterization model is established.

Calculate the degree of composition of each index state, and then obtain the single-factor fuzzy discrimination matrix of the entire system, as shown below.

$$R = \begin{bmatrix} \mu_1^1 & \mu_1^2 & \mu_1^3 & \mu_1^4 \\ \mu_2^1 & \mu_2^2 & \mu_2^3 & \mu_2^4 \\ \mu_3^1 & \mu_3^2 & \mu_3^3 & \mu_3^4 \end{bmatrix} \tag{4}$$

Among them, the μ_1^1 strongly transitive fuzzy matrix reflects the membership of each traffic state level of each evaluation index pair corresponding to the calculated value of the membership function.

Determining weights Describe the importance of each indicator according to the traffic situation. It is necessary to determine the weight of the basic indicators. Each indicator weight constitutes a weight set, namely:

$$\omega = (\omega_1, \omega_2, \omega_3) \tag{5}$$

Establish an evaluation system-construct a judgment matrix for comparison According to the 9-scale method, the importance between n elements of the same layer can be obtained, thereby establishing a judgment matrix.

$$I = \begin{bmatrix} W_{11} & \cdots & W_{1n} \\ \vdots & \ddots & \vdots \\ W_{m1} & \cdots & W_{mn} \end{bmatrix} \tag{6}$$

The elements of the judgment matrix are normalized. The formula is as follows:

$$\alpha_{ij} = \frac{\alpha_{ij}}{\sum_{i=1}^{n} \alpha_{ij}} \tag{7}$$

n——Index factor number of each column judgment matrix

The judgment matrix generated after normalization is added row by row:

$$w_i = \sum_{i=1}^{n} a_{ij} \tag{8}$$

In——The number of index factors for each judgment matrix

Based on the judgment matrix obtained above, normalization is performed to obtain a maximum feature vector, which is the weight of each factor. $w = [w_1, w_2, w_3 \ldots, w_n]$. For a multi-level evaluation system, the weights of the upper layer indicators of each factor indicator can be determined from top to bottom, and finally the weight of each layer factor relative to the target layer is obtained.

Fuzzy comprehensive evaluation, after determining the single-factor discriminant matrix of evaluation index weights, comprehensive evaluation can be conducted through fuzzy transformation.

$$B = (\mathrm{w}_1, \mathrm{w}_2, \mathrm{w}_3)^{\circ}\begin{bmatrix} \mu_1^1 & \mu_1^2 & \mu_1^3 & \mu_1^4 \\ \mu_2^1 & \mu_2^2 & \mu_2^3 & \mu_2^4 \\ \mu_3^1 & \mu_3^2 & \mu_3^3 & \mu_3^4 \end{bmatrix} = \left(\mathrm{b}_1, \mathrm{b}_2, \mathrm{b}_3, \mathrm{b}_4\right) \tag{9}$$

$$\mathrm{b}_\mathrm{j} = \sum_{\mathrm{i}=1}^{3} \omega_\mathrm{i}\mu_\mathrm{i}^\mathrm{j}, \mathrm{j} = 1, 2, 3, 4 \tag{10}$$

b_j——For the final fuzzy comprehensive evaluation index, comprehensively consider the membership degree of all basic indexes in the evaluation set to the jth element. According to the principle of maximum membership, the evaluation element with the largest membership is selected as the evaluation result.

3.4 Classification of Congestion Grade Based on Fuzzy Comprehensive Evaluation

According to the calculation method of traffic congestion in Shenzhen issued by the state in 2011. It quantifies the traffic condition based on the average speed of the statistical interval. The value range is 0–100. Eventually, four traffic congestion states of Smoother, Smooth, Crowded and Blockage. The basic calculation model and the weighted calculation model are suitable for the traffic state assessment of road segments and road networks, respectively. The specific method is as follows:

The key parameter of the TSI calculation model is the formation speed, which is used to evaluate the traffic state of the road section. The formula is as follows:

$$TSI = \frac{\mathrm{V}_\mathrm{f} - \mathrm{V}_\mathrm{i}}{\mathrm{V}_\mathrm{f}} \times 100 \tag{11}$$

V_f——Study actual road speed

V_i——Study free flow speed

Quantify traffic conditions based on the average speed of the statistical interval. The value range is 0–100, which divides the traffic operation status into very stable, stable, congested and congested. The table shows the correspondence between TSI and road network traffic (Table 1).

Table 1. Division of TSI

Traffic status level	Smoother	Smooth	Crowded	Blockage
TSI	[0, 30)	[30, 50)	[50, 70)	[70, 100]

The basic calculation model is relatively simple, and the results can be calculated from the average stroke speed and free flow speed of the link. However, in practical applications, the evaluation of the traffic congestion status of road sections is not enough, and the traffic status of the road network needs to be carried out. Therefore, considering the influence of the number of road segments and miles on the degree of congestion, the formula for the weighted calculation model is as follows.

$$TSI = \frac{\sum_{i=1}^{l} K_i l_i \left[\frac{V_f - V_i}{V_f}\right]}{\sum_{i=1}^{l} K_i} \times 100 \tag{12}$$

K_i——Lane number i

l_i—Section i Road length, *km.*

l——Total number of road segments

After obtaining the road network TSI, convert it to a traffic status level according to the table. According to the comparison of TSI road travel time, Shenzhen looks at the traffic status of the road network.

3.5 Comparison of Various Algorithm Effects

Through the selected traffic congestion measurement indicators, and then use random forest, adaboost, GBDT, LassoCV, BP neural network algorithms to train and test the data, and compare the effectiveness of various algorithms in dealing with the problem.

3.6 BP Neural Network Regression Prediction

Through the "training" to obtain this input, the appropriate nonlinear relationship between the output. The "training" process can be divided into two stages of forward transmission and backward transmission:

Forward transmission phase:
Take a sample from the sample set P_i, Q_i; Enter P_i into the network; Calculate the error measure E_i and the actual output $O_i = F_L(\ldots(F_2(F_1(P_i W^{(1)})W^{(2)}))W^{(L)})$; Make adjustments to the weight values $W^{(1)}$ ....$W^{(2)} \ldots W^{L}$, and repeat this cycle until $\sum E_i < \varepsilon$.
Backward propagation stage-error propagation stage:
Calculate the difference between the actual output O_P and the ideal output Q_i; Adjust the weight matrix of the output layer with the error of the output layer.

$$E_i = \frac{1}{2}\sum_{j=1}^{m}(Q_{ij} - O_{ij})^2 \tag{13}$$

Use this error to estimate the error of the direct leader layer of the output layer, and then use the error estimation of the output layer leader layer. Change the error of the previous layer. In this way, the error estimates of all other layers are obtained. And use these estimates to modify the weight matrix. Form a process of gradually transmitting the error displayed at the output to the output in the opposite direction to the output signal.

The error measure of the network about the entire sample set:

$$E = \sum E_I \tag{14}$$

4 Experiment

4.1 Index Correlation Coefficient

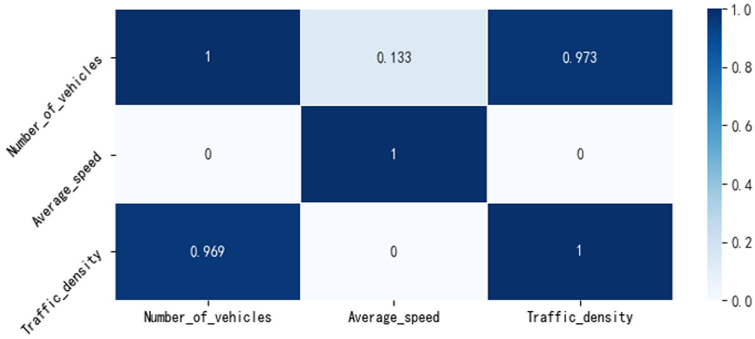


Fig. 2. Correlation coefficients of various indicators

As shown in Fig. 2, the c or relationship coefficient of the number of vehicles, average speed and traffic density extracted according to data processing. It is known from the correlation that although the number of taxis is extracted, there is no difference between the average speed of vehicles on the same road segment and the average speed of non-rented vehicles, so the calculated traffic after observing the extracted data The trend of density change is still representative on the road section.

4.2 Analysis of Congestion in Fuzzy Comprehensive Evaluation

The following figure is obtained by analyzing the relationship between traffic flow and traffic flow density.

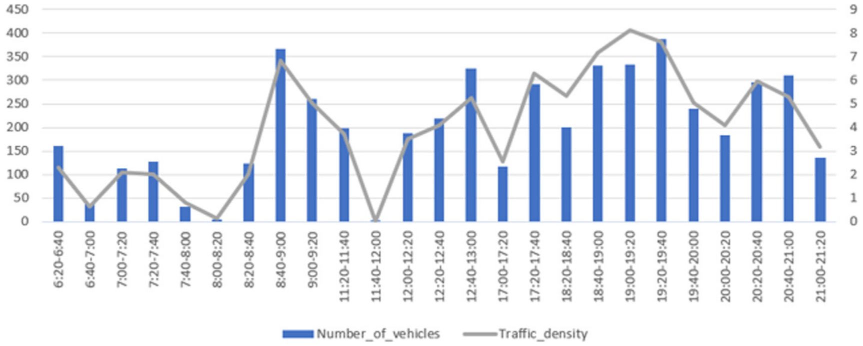


Fig. 3. Relationship between traffic flow and traffic flow density

In the relationship between the number of vehicles and the traffic density in Fig. 3, it can be found that the early, middle, and late high stages are generally between 8:00–9:00, 12:00–1:00, and 19:00–20:00 (Table 2).

Table 2. Congestion level and time prediction during congestion period

Period of time	Number of vehicles	Congestion level	Expected congestion time (min)
8:00–8:20	5	Soomther	5–10
8:20–8:40	124	Crowded	10–20
8:40–9:00	367	Blockage	20–30
12:00–12:20	188	Blockage	20–35
12:20–12:40	218	Blockage	35–50
12:40–13:00	324	Blockage	30–40
19:00–19:20	334	Blockage	35–50
19:20–19:40	184	Blockage	35–55
19:40–20:00	136	Blockage	25–35

Through fuzzy comprehensive evaluation, the degree of congestion is divided into 4 levels, and the analysis of the congestion in each time period is shown in the figure above. It was found that the congestion situation was the most serious in the morning, middle and evening peak congestion time within a day at 8:40–9:00, 12:40–1:00, 19:20–19:40, and the number of vehicles coming out of GPS vehicles was 367, 324, 388, providing conditions for future short-term traffic forecast.

4.3 Comparison of Various Algorithms

We used Anaconda3's Jupiter-notebook software and called the linear model. Lasso CV algorithm module corresponding to the sklearn library and the Random Forest Regressor, Ada Boost Regressor, Gradient Boosting Regressor algorithm modules corresponding to the sklearn. ensemble library, and the Grid Search CV model regulator corresponding to sklearn. model selection Then each algorithm was trained and adjusted, and the relationship between the obtained test set and the real value is shown in Fig. 4.

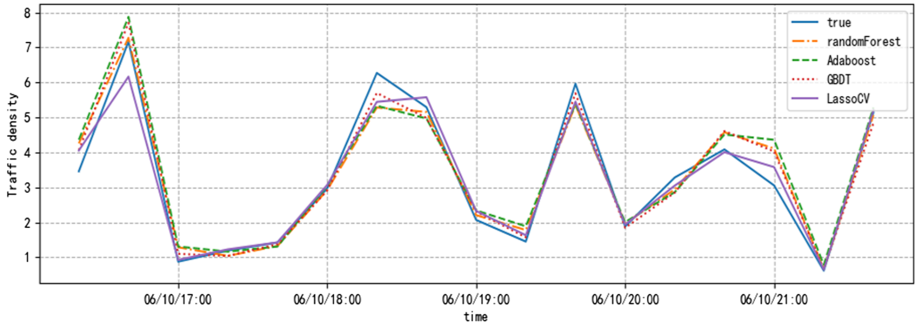


Fig. 4. Comparison of accuracy of various algorithms

As can be seen from Fig. 4, various algorithms have a good effect on traffic congestion analysis, but there are still deviations in certain data. Therefore, we use Anaconda's jupyter notebook software, call the tensorflow module to implement the BP neural network model, and train and test the data. The relationship between the obtained test results and the true value is shown in Fig. 5.

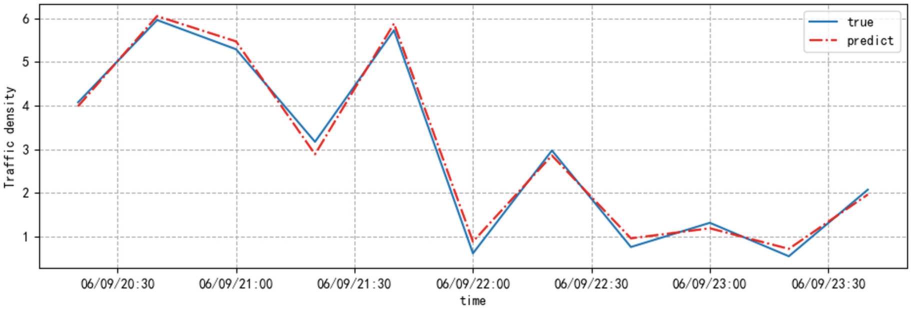


Fig. 5. BP neural network accuracy chart

It can be seen from the prediction results of the BP neural network that the algorithm has a significant deviation in processing a small part of the data in processing this problem, and the overall effect is better. The accuracy of its various algorithms and the optimal parameters for tuning are as follows.

Table 3. Comparison table of accuracy of multiple models

Model	Mean squared error	Hyperparameter
Random forest regression	0.1195	'n_estimators':920
Adaboost regression	0.2127	'n_estimators':1750
GBDT regression	0.1654	'n_estimators':570
Lasso regression	0.1309	Kernel='linear'
BP neural network	0.0190	Number of hidden layer nodes:5

It can be seen from Table 3 that various algorithms have better effects on this problem after training and adjustment. Among them, the BP neural network has the best effect, and the mean squared error has reached 0.0190.

4.4 BP Neural Network Prediction

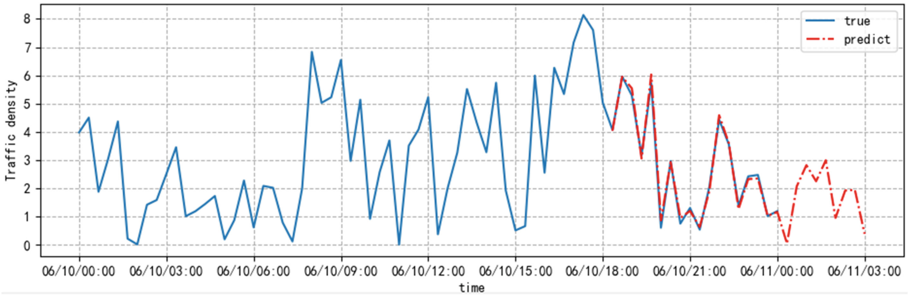


Fig. 6. BP neural network predicts the future time period diagram

By using the data from June 9 to train and test the BP neural network, the mean square error (MSE) of the model is 0.01905, which is relatively reliable. Then predict the congestion of the road within three hours. It can be seen that during the time period from 00:00 to 03:00 on June 10, the road congestion will maintain a small fluctuation trend, but the overall amplitude does not change significantly, that is, the road will not appear serious in the short term Congestion. In addition, you can use the above method-fuzzy comprehensive evaluation in this article to process the prediction results, that is, you can get the congestion situation and congestion time in this period (Fig. 6).

5 Conclusion

It can be seen from the above analysis that the various algorithms after parameter optimization are consistent with the short-term traffic prediction results and recorded values, and the results obtained by these algorithms are all valid. However, the prediction result of BP neural network is more accurate than other algorithms, which shows that BP neural network has stronger data expression ability and can better simulate short-term traffic conditions. Within the acceptable error range, the prediction results can provide effective decision-making information for emergency rescue of urban center road accidents. In the future 5G mobile network, cars with GPS positioning system can accurately analyze the current location of the vehicle and the traffic flow of the next section and the surrounding sections that will pass. Immediately before entering the traffic jam section, the most accurate algorithm is used to recommend a new relatively smooth section for the vehicle. On this basis, the number of vehicles in each section is regulated in a balanced manner to prevent the congestion from further deteriorating, and at the same time provide a guarantee for citizens to travel smoothly and efficiently, laying a foundation for the development of intelligent traffic control systems. Although with the increase of traffic congestion, the prediction results of the model may be deviated to a certain extent, and various traffic parameters will change with time, but it can ensure that the driver occurs in a traffic accident or bad weather. The accident allows the command center to give a rescue plan based on the traffic situation that occurred within a short period of time and arrive at the scene as soon as possible.

References

1. Danqing, K.: Research on short-term traffic flow prediction method based on deep learning. Harbin University of Science and Technology, Heilongjiang, pp. 7714–2015 (2018)
2. Liu, Z., Du, Y., Yan, D., et al.: Short-term traffic flow prediction based on the combination of K-nearest neighbor algorithm and support vector regression. Highway Traff. Sci. Technol. **34**(5), 122–128 (2017)
3. Fu, G., Qiang, K., Lu, F., et al.: Short-term traffic flow prediction model based on support vector machine regression. J. South China Univ. Technol. (Nat. Sci. Ed.) **41**(9), 71–76 (2013)
4. Yan, Y., Bai, L, Wu, Q., et al.: Traffic congestion prediction and evaluation based on multi-index fuzzy comprehensive evaluation. Comput. Appl. Res. **36**(12), 3697–3700, 3704 (2019)
5. Zhu, B., Fu, Z., Yang, S., et al.: Forecasting model of traffic accident spatio-temporal influence based on nonlinear regression and BP neural network. Highway Eng. **43**(6), 134–139 (2018)
6. Li, Y., Liu, L., Wang, Y.: Short-term traffic flow prediction based on combined prediction model. Transp. Syst. Eng. Inf. **13**(2), 34–41 (2013)

Analysis of Breast Cancer Detection Using Different Machine Learning Techniques

Siham A. Mohammed[1], Sadeq Darrab[3](✉), Salah A. Noaman[2], and Gunter Saake[3]

[1] Taiz University, Taiz, Yemen
siham.alkadasi@gmail.com
[2] Aden University, Aden, Yemen
s-salah-17@hotmail.com
[3] University of Magdeburg, Magdeburg, Germany
{sadeq.darrab,gunter.saake}@ovgu.de

Abstract. Data mining algorithms play an important role in the prediction of early-stage breast cancer. In this paper, we propose an approach that improves the accuracy and enhances the performance of three different classifiers: Decision Tree (J48), Naïve Bayes (NB), and Sequential Minimal Optimization (SMO). We also validate and compare the classifiers on two benchmark datasets: Wisconsin Breast Cancer (WBC) and Breast Cancer dataset. Data with imbalanced classes are a big problem in the classification phase since the probability of instances belonging to the majority class is significantly high, the algorithms are much more likely to classify new observations to the majority class. We address such problem in this work. We use the data level approach which consists of resampling the data in order to mitigate the effect caused by class imbalance. For evaluation, 10 fold cross-validation is performed. The efficiency of each classifier is assessed in terms of true positive, false positive, Roc curve, standard deviation (Std), and accuracy (AC). Experiments show that using a resample filter enhances the classifier's performance where SMO outperforms others in the WBC dataset and J48 is superior to others in the Breast Cancer dataset.

Keywords: Breast cancer · Classification · Data mining

1 Introduction

Breast cancer is the second leading cause of death among women worldwide [1]. In 2019, 268,600 new cases of invasive breast cancer were expected to be diagnosed in women in the U.S., along with 62,930 new cases of non-invasive breast cancer [2]. Early detection is the best way to increase the chance of treatment and survivability. Data mining has become a popular tool for knowledge discovery which shows good results in marketing, social science, finance and medicine [19, 20]. Recently, multiple classifiers algorithms are applied on medical datasets to perform predictive analysis about patients and their medical diagnosis [6, 9, 10, 21]. For example, using machine learning techniques to assess tumor behavior for breast cancer patients. One problem is that there is a class

Y. Tan et al. (Eds.): DMBD 2020, CCIS 1234, pp. 108–117, 2020.
https://doi.org/10.1007/978-981-15-7205-0_10

imbalance in the training data, since the probability of not having this disease is higher than the one of having it. This paper introduces a comparison between three different classifiers: J48, NB, and SMO with respect to accuracy in detection of breast cancer. Our aim is to prepare the dataset by proposing a suitable method that can manage the imbalanced dataset and the missing values, to enhance the classifier's performance. All tasks were conducted using Weka 3.8.3.

The remainder of this paper is organized as follows. Section 2 presents literature review. Section 3 introduces the datasets. Section 4 describes the research methodology including pre-processing experiments, classification and performance evaluation criteria. The experimental results are presented in Sect. 5. Finally, Sect. 6 shows the conclusion and future work.

2 Literature Review

In recent years, several studies have applied data mining algorithms on different medical datasets to classify Breast Cancer. These algorithms show good classification results and encourage many researchers to apply these kind of algorithms to solve challenging tasks. In [21], a convolutional neural network (CNN) was used to predict and classify the invasive ductal carcinoma in breast histology images with an accuracy of almost 88%. Moreover, data mining is used widely in medical fields to predict and classify abnormal events to create a better understanding of any incurable diseases such as cancer. The outcomes of using data mining in classification are promising for breast cancer detection. Therefore, data mining approach is used in this work. A list of some literature studies related to this method is presented in Table 1.

3 Datasets

The datasets that are used in this paper are available at the UCI Machine Learning Repository [13].

3.1 WBC Dataset

The WBC dataset contains 699 instances and 11 attributes in which 458 were benign and 241 were malignant cases [14]. In the WBC, the value of the attribute (Bare Nuclei) status was missing for 16 records. Hence data preprocessing is essential and important for this dataset, requiring us to manage the imbalanced data and the missing values.

3.2 Breast Cancer Dataset

The feature form this dataset are computed from a digitized image of a fine needle aspirate (FNA) of a breast tumor. The target feature records the prognosis (i.e., malignant or benign). The dataset contains 286 instances and 10 attributes in which 201 were no-recurrence-events and 85 were recurrence events. In the Breast Cancer dataset, the value of the attribute (node-caps) status was missing in 8 records.

Table 1. Breast cancer detection research using different machine learning algorithms.

Paper title	Datasets	Algorithms	Results
Integration of data mining classification techniques and ensemble learning for predicting the type of breast cancer recurrence [3], 2019	Breast Cancer	NB, SVM, GRNN and J48	GRNN & J48 accuracy: 91% NB & SVM: 89%
A study on prediction of breast cancer recurrence using data mining techniques [4], 2017	WPBC	Classification: KNN, SVM, NB and C5.0, Clustering: K-means, EM, PAM and Fuzzy c-means	Classification accuracy is better than clustering, SVM & C5.0: 81%
Predicting breast cancer recurrence using effective classification and feature selection technique [5], 2016	WPBM	NB, C4.5, SVM	NB: 67.17%, C4.5: 73.73%, SVM: 75.75%
Using machine learning algorithms for breast cancer risk prediction and diagnosis [6], 2016	WBC	SVM, C4.5, NB, KNN	SVM outperform others: 97.13%
Study and analysis of breast cancer cell detection using Naïve Bayes, SVM and ensemble algorithms [7], 2016	WDBC	NB, SVM, Ensemble	SVM: 98.5%, NB & Ensemble: 97.3%
Analysis of Wisconsin breast cancer dataset and machine learning for breast cancer detection [8], 2015	WDBC	NB, J48	NB: 97.51%, J48: 96.5%
Comparative study on different classification techniques for breast cancer dataset [9], 2014	Breast Cancer	J48, MLP, rough set	J48: 79.97%, MLP: 75.35%, rough set: 71.36%
A novel approach for breast cancer detection using data mining techniques [10], 2014	WBC	SMO, IBK, BF Tree	SMO: 96.19%, IBK: 95.90%, BF Tree: 95.46%
Experiment comparison of classification for breast cancer diagnosis [11], 2012	WBC WDBC WPBC	J48, SMO, MLP, NB, IBK	In WBC: MLP & J48: 97.2818%. In WDBC: SMO: 97.7% or fusion on SMO & MLP: 97.7% In WPBC: fusion of MLP, J48, SMO and IBK: 77%
Analysis of feature selection with classification: breast cancer datasets [12], 2011	WBC WDBC Breast Cancer	Decision Tree with and without feature selection	Feature selection enhances the results WBC: 96.99% WDBC: 94.77% Breast Cancer: 71.32%

4 Research Methodology

The two datasets used in this work are vulnerable to missing and imbalanced data therefore, before performing the experiments, a large fraction of this work will be for preprocessing the data in order to enhance the classifier's performance. Preprocessing will focus on managing the missing values and the imbalanced data. To manage the missing attributes, all the instances with missing values are removed. The imbalance data problem needs to adjust either the classifier or the training set balance. To do so, the resample filter is used to rebalance the data artificially. Then, 10 fold cross validation is applied and finally a comparison between these three classifiers is implemented.

4.1 Preprocessing Phase

First, the data were discretized using discretize filter, then missing values were removed from the dataset. Second, instances were resampled using the resample filter in order to maintain the class distribution in the subsample and to bias the class distribution toward a uniform distribution. Section 5 will show that this idea is improving the classifier's performance. Third, 10 fold cross validation was applied then experiments were carried out over three classifiers Naïve Bayes, SMO and J48, as illustrated in Fig. 1.

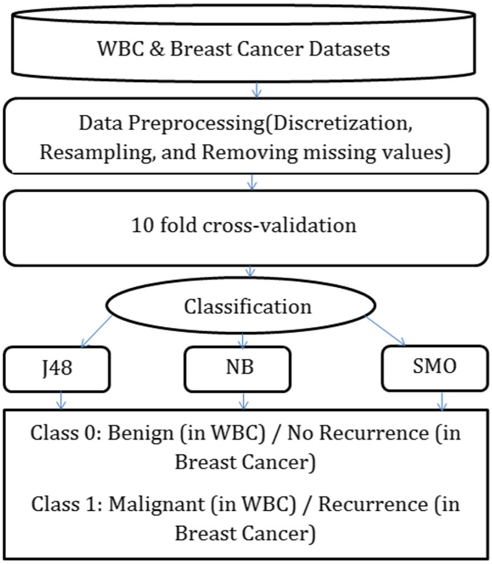

WBC & Breast Cancer Datasets
Data Preprocessing(Discretization, Resampling, and Removing missing values)
10 fold cross-validation
Classification
J48
NB
SMO
Class 0: Benign (in WBC) / No Recurrence (in Breast Cancer)
Class 1: Malignant (in WBC) / Recurrence (in Breast Cancer)

Fig. 1. Proposed breast cancer detection model using Breast Cancer and WBC datasets.

In Fig. 1, the data preprocessing technique has been applied including three steps: discretization, instances resampling and removing the missing values. After that, 10 fold cross validation has been applied. Then, three classifiers have been evaluated over the prepared datasets.

4.2 Training and Classification

In order to minimize the bias associated with the random sampling of the training data, we use 10 fold cross validation after the pre-processing phase. In k-fold cross-validation, the original dataset is randomly partitioned into k equal size subsets. The classification model is trained and tested k times. Each time, a single subset is retained as the validation data for testing the model, and the remaining k−1 subsets are used as training data. Three classification techniques were selected: a Naïve Bayes (NB), a Decision Tree built on the J48 algorithm, and a Sequential Minimal Optimization (SMO). The NB classifier is a probabilistic classifier based on the Bayes rule. It works by estimating the portability of each class value that a given instance belongs to that class [15]. The J48 algorithm [16] uses the concept of information entropy and works by splitting each data attributers into smaller datasets in order to examine entropy differences. It is an improved and enhanced version of C4.5 [17]. The SMO model implements John Platt's sequential minimal optimization algorithm for training a support vector classifiers. This implementation globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default [18].

4.3 Performance Evaluation Criteria

In this study, we use five performance measures to evaluate all the classifiers: true positive, false positive, ROC curve, standard deviation (Std) and accuracy (AC).

$$AC = (TP + TN)/(TP + TN + FP + FN). \tag{1}$$

Where TP, TN, FP and FN denote true positive, true negative, false positive and false negative, respectively.

5 Experimental Results

First, the three classifications algorithms were tested on the WBC and the Breast Cancer datasets without applying the preprocessing techniques. Among them, the best result was recorded for J48: 75.52% in the Breast Cancer dataset and for SMO: 96.99% in the WBC dataset. Next, after applying preprocessing techniques accuracy increases to 98.20% with J48 in the Breast Cancer dataset and 99.56% with SMO in the WBC dataset.

5.1 Experiment Using the Breast Cancer Dataset

First, the three classifiers are tested over original data (without any preprocessing).The results show that J48 is the best one with 75.52% accuracy where the accuracy of NB and SMO are 71.67% and 69.58%, respectively. Next, we apply discretization filter and remove the records with missing values, results improved with NB and SMO as follows: NB: 75.53% and SMO: 72.66% where J48: 74.82%. After that, resample filter was applied for 7 times. The Performance of the classifiers are improved and enhanced as shown in Table 2.

Table 2. Performance of the classifiers in the Breast Cancer Dataset.

Experiments steps	Classifier accuracy		
	J48	NB	SMO
Original without preprocessing	75.52%	71.67%	69.58%
After removing missing values & discretization	74.82%	75.53%	72.66%
After applying resample filter (first time)	79.49%	77.33%	80.93%
Applying resample filter (second time)	81.65%	78.05%	80.57%
Applying resample filter (third time)	87.41%	78.41%	82.73%
Applying resample filter (fourth time)	92.08%	77.69%	88.84%
Applying resample filter (fifth time)	95.68%	79.13%	91.72%
Applying resample filter (sixth time)	97.48%	79.85%	95.68%
Applying resample filter (seventh time)	98.20%	76.61%	95.32%

As illustrated in Table 2, we can obviously notice that the more resample filter we apply, the improved accuracy we obtain. That is because the data is imbalanced and the filter maintains the class distribution. For the Breast cancer dataset, J48 outperforms others with 98.20%. Accuracy measures for J48 classifier is shown in Table 3 and Roc curve of J48 is shown in Fig. 2.

Table 3. Accuracy measures for J48 in the Breast Cancer Dataset.

TP	FP	Precision	Recall	Roc curve	Std	Class
1.000	0.049	0.980	0.996	1.000	0.5678	No-recurrence-events
0.951	0.000	1.000	0.996	0.951		Recurrence-events

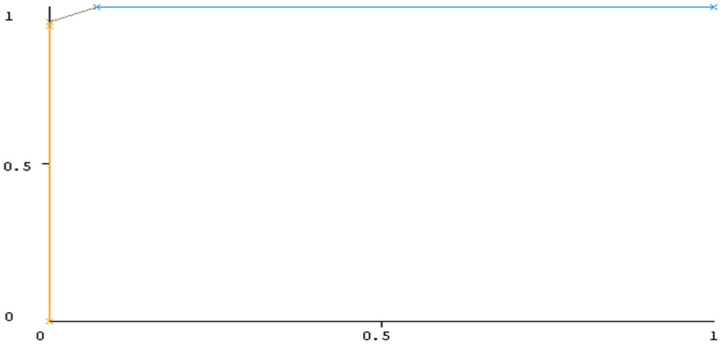


Fig. 2. J48 ROC curve in Breast Cancer Dataset.

To measure the performance of the proposed model, we compare the obtained results with the study proposed in [9]. The same dataset and three classifiers including J48 algorithm are used to evaluate the model's performance. According to the results, the J48 classifier of the proposed model achieves high accuracy comparing to other classifiers. This is because of using the resample filter for the pre-processing phase in the proposed model rather than feature selection technique that used in [9] as illustrated in Table 4.

Table 4. Compression of accuracy measures for the Breast Cancer Dataset.

Methodology	Study [9]	Proposed method
With out pre-processing	None	J48: 75.52%, NB: 71.67% SMO: 69.58%
With pre-processing	Missing values were replaced with WEKA pre-processing techniques and feature selection was applied J48: 79.97%, MLP: 75.35% & rough set: 71.36%	Delete records of missing values and Descretization J48: 74.82%, NB: 75.53% SMO: 72.66%
Using the resample filter	None	Applying the resample filter for 7 times J48: 98.20%, NB: 76.61% SMO: 95.32%

5.2 Experiment Using the WBC Dataset

Same experiments were applied with the WBC dataset. With respect to applying preprocessing techniques all algorithms present higher classification accuracy, the difference lies in the fact that using the resample filter several times improves the classification accuracy. SMO classifier achieve 99.56% efficiency compared to 99.12% of the Naïve Bayes and 99.24% of the J48. Results are illustrated in Table 5.

Table 5. Performance of the classifiers in WBC dataset.

Experiments steps	Classifier accuracy		
	J48	NB	SMO
Original Without preprocessing	94.56%	95.99%	96.99%
After removing missing values & discretization	95.91%	97.37%	96.78%
After applying resample filter (first time)	95.91%	97.51%	98.97%
Applying resample filter (second time)	97.95%	98.10%	99.41%
Applying resample filter (third time)	98.68%	98.10%	99.12%
Applying resample filter (fourth time)	99.24%	99.12%	99.56%

In the WBC dataset, SMO superior than others with 99.56%. Accuracy measures for SMO classifier is shown in Table 6 and Roc curve of SMO is shown in Fig. 3.

Table 6. Accuracy measures for SMO in WBC Dataset.

TP	FP	Precision	Recall	Roc curve	Std	Class
0.996	0.004	0.998	0.996	0.996	0.2220	Benign
0.996	0.004	0.992	0.996	0.996		Malignant

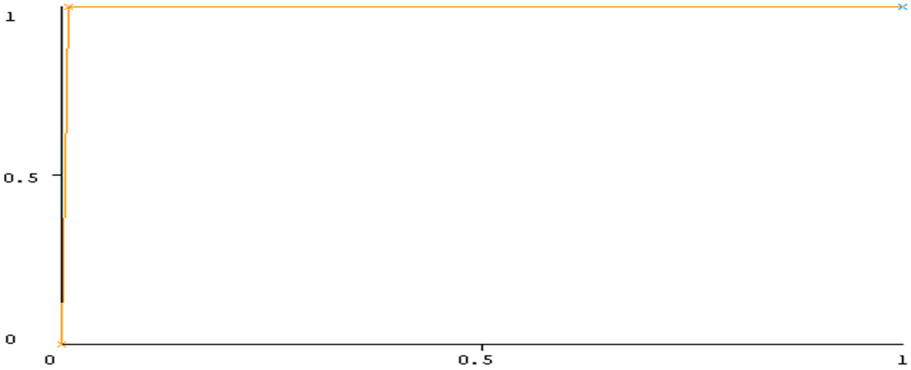


Fig. 3. SMO ROC curve in WBC Dataset.

In terms of the WBC dataset, our proposed method is compared with two studies [6, 10]. Results shows that the performance of SMO classifier is better since our model employs pre-processing, and resampling approaches. Thus, utilizing pre-processing, and resampling techniques play an important role in increasing the SMO accuracy comparable to the other techniques in [6, 10]. Details are shown below in Table 7.

Table 7. Compression of accuracy measures for the WBC Dataset.

Methodology	Study [6]	Study [10]	Proposed method
Without pre-processing	C4.5: 95% NB: 95.9% SVM: 97.3%	SMO: 96.19%, IBK: 95.90%, BF Tree: 95.46%	J48: 94.56%, NB: 95.99% SMO: 96.99%
With pre-processing	None	None	Delete records of missing values and Descretization J48: 95.91%, NB: 97.37% and SMO: 96.78%
Using the resample filter	None	None	Applying the resample filter for 4 times J48: 99.24%, NB: 99.12%, SMO: 99.56%

6 Conclusion

Breast cancer is considered to be one of the significant causes of death in women. Early detection of breast cancer plays an essential role to save women's life. Breast cancer detection can be done with the help of modern machine learning algorithms. In this paper, we focus on how to deal with imbalanced data that have missing values using resampling techniques to enhance the classification accuracy of detecting breast cancer. In our work, three classifiers algorithms J48, NB, and SMO applied on two different breast cancer datasets. Results show that using the resample filter in the preprocessing phase enhances the classifier's performance. In the future, the same experiments will apply to different classifiers and different datasets.

References

1. U.S. Cancer Statistics Working Group. United States Cancer Statistics: 1999–2008 Incidence and Mortality Web-based Report. Atlanta (GA): Department of Health and Human Services, Centers for Disease Control
2. http://www.breastcancer.org/symptoms/understand_bc/statistics
3. Silva, J., Lezama, O.B.P., Varela, N., Borrero, L.A.: Integration of data mining classification techniques and ensemble learning for predicting the type of breast cancer recurrence. In: Miani, R., Camargos, L., Zarpelão, B., Rosas, E., Pasquini, R. (eds.) GPC 2019. LNCS, vol. 11484, pp. 18–30. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19223-5_2
4. Ojha U., Goel, S.: A study on prediction of breast cancer recurrence using data mining techniques. In: 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence, IEEE, pp. 527–530, 2017
5. Pritom, A.I., Munshi, M.A.R., Sabab, S.A., Shihab, S.: Predicting breast cancer recurrence using effective classification and feature selection technique. In: 19th International Conference on Computer and Information Technology (ICCIT), pp. 310–314. IEEE (2016)
6. Asri, H., Mousannif, H., Al, M.H., Noel, T.: Using machine learning algorithms for breast cancer risk prediction and diagnosis. Procedia Comput. Sci. **83**, 1064–1069 (2016)
7. Hazra, A., Mandal, S.K., Gupta, A.: Study and analysis of breast cancer cell detection using Naïve Bayes, SVM and Ensemble Algorithms. Int. J. Comput. Appl. **145**, 0975–8887 (2016)
8. Rodrigues, B.L.: Analysis of the Wisconsin Breast Cancer dataset and machine learning for breast cancer detection. In: Proceedings of XI Workshop de Visão Computational, pp. 15–19 (2015)
9. Saabith, A.L.S., Sundararajan, E., Bakar, A.A.: Comparative study on different classification techniques for breast cancer dataset. Int. J. Comput. Sc. Mob. Comput. **3**(10), 185–191 (2014)
10. Chaurasia, V., Pal, S.: A novel approach for breast cancer detection using data mining techniques. Int. J. Innovative Res. Comput. Commun. Eng. **2** (2017). (An ISO 3297: 2007 Certified Organization)
11. Salama G.I., Abdelhalim, M.B., Zeid, M.A.E.: Experimental comparison of classifiers for breast cancer diagnosis. In: 2012 Seventh International Conference on Computer Engineering & Systems (ICCES), pp. 180–185. IEEE (2012)
12. Lavanya, D., Rani, D.K.U.: Analysis of feature selection with classification: breast cancer datasets. Indian J. Comput. Sci. Eng. (IJCSE), pp. 756–763 (2011)
13. Breast Cancer Wisconsin Dataset. Available at: UCI Machine Learning Repository
14. Dataset Description. Available at: UCI Machine Learning Repository
15. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques. Elsevier, New York (2011)

16. Quinlan, R.C.: 4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
17. Quinlan, J.R.: Simplifying decision trees. Int. J. Man-Mach. Stud. **27**, 221–234 (1987)
18. Piatt, J.: Fast training of support vector machines using sequential minimal optimization. In: Advances in Kernel Methods-Support Vector Learning (1998)
19. Darrab, S., Ergenc, B., Vertical pattern mining algorithm for multiple support thresholds. In: International Conference on Knowledge Based and Intelligent Information and Engineering (KES), Procedia Computer Science, vol. 112, pp. 417–426 (2017)
20. Darrab, S., Ergenc, B.: Frequent pattern mining under multiple support thresholds, the International Conference on Applied Computer Science (ACS). Wseas Transactions on Computer Research, pp. 1–10 (2016)
21. Alghodhaifi, H., Alghodhaifi, A., Alghodhaifi, M.: Predicting Invasive Ductal Carcinoma in breast histology images using Convolutional Neural Network. In: 2019 IEEE National Aerospace and Electronics Conference (NAECON), pp. 374–378 (2019)

Author Index