P. Krishna Reddy
Ashish Sureka
Sharma Chakravarthy
Subhash Bhalla (Eds.)

LNCS 10721

Big Data Analytics

5th International Conference, BDA 2017
Hyderabad, India, December 12–15, 2017
Proceedings

Springer

Lecture Notes in Computer Science 10721

More information about this series at http://www.springer.com/series/7409

P. Krishna Reddy · Ashish Sureka
Sharma Chakravarthy · Subhash Bhalla (Eds.)

Big Data Analytics

5th International Conference, BDA 2017
Hyderabad, India, December 12–15, 2017
Proceedings

Springer

Editors
P. Krishna Reddy
International Institute
of Information Technology
Hyderabad
India

Ashish Sureka
Rajiv Gandhi Education City
Sonepat
India

Sharma Chakravarthy
University of Texas at Arlington
Arlington, TX
USA

Subhash Bhalla
University of Aizu
Aizu-Wakamatsu
Japan

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-72412-6 ISBN 978-3-319-72413-3 (eBook)
https://doi.org/10.1007/978-3-319-72413-3

Library of Congress Control Number: 2017960961

LNCS Sublibrary: SL3 – Information Systems and Applications, incl. Internet/Web, and HCI

Printed on acid-free paper

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

Preface

Business data analytics in scientific domains depends on computing infrastructure. A scientific exploration of data is beneficial for large-scale public utility services, either directly or indirectly. Many research efforts are being made in diverse areas, such as big data analytics and cloud computing, sensor networks, and high-level user interfaces for information accesses by common users. Government agencies in many countries plan to launch facilities in education, health care, and information support as a part of e-government initiatives. In this context, big data analytics and information interchange management have become active research fields. A number of new opportunities have evolved in design and modeling based on the new computing needs of users. Database systems play a central role in supporting networked information systems for access and storage management aspects.

The 5th International Conference on Big Data Analytics (BDA) 2017 was held during December 12–15 at the International Institute of Information Technology (IIIT), Hyderabad. The program included research contributions and invited contributions. A view of research activity in big data analytics, information interchange management, and associated research issues was provided by the sessions on related topics. The keynote address was contributed by Prof. Krithi Ramamritham for the session on big data analytics. Additional sessions were organized to cover information and knowledge management, mining of massive datasets, conceptual modeling, and data mining and analysis. I would like to thank the members of the Program Committee for their support and all authors who considered BDA 2017 in making research contributions. Within the big data analytics framework, the conference attracted submissions under diverse topics, such as analysis and prediction (graphs, stock markets, tweets, fraud), machine-learning approaches for intrusion detection, time-series and text data, and applications (health care, news, and agriculture) indicating the importance of this growing area from the research and application perspective. The selected papers in these proceedings, along with keynotes and tutorials on a variety of relevant topics, are expected to further stimulate research and exposure to cutting-edge research.

The conference received 80 submissions. Each was reviewed by at least three Program Committee members. A few papers had four reviews to get an additional opinion for final decision-making. The committee selected ten papers. Three out of ten papers were conditionally accepted. Authors of such papers went through a shepherding process to improve their papers. Finally, the acceptance rate was 12.5%. The Program Committee comprised 45 members from 10 countries. Papers were received from authors from seven different countries. The authors of accepted papers were from three countries.

The sponsoring organizations, the Steering Committee, and the Organizing Committee deserve praise for the support they provided. A number of individuals contributed to the success of the conference. I thank Prof. Krithi Ramamritham for providing continuous support and encouragement.

The conference received invaluable support from IIIT Hyderabad. In this context, I thank Prof. P. J. Narayanan, Director IIIT Hyderabad. Many thanks are also extended to the faculty members at the institute for their cooperation and support.

December 2017

P. Krishna Reddy
Ashish Sureka
Sharma Chakravarthy
Subhash Bhalla

Organization

BDA 2017 was organized by the Data Science and Analytics Center, Kohli Center on Intelligent Systems (KCIS), International Institute of Information Technology Gachibowli, Hyderabad, India.

Honorary Chairs

U. B. Desai	IIT Hyderabad, India
P. J. Narayanan	IIIT Hyderabad, India

General Chair

P. Krishna Reddy	IIIT Hyderabad, India

Steering Committee

Krithi Ramamritham	IIT Bombay, India
S. K. Gupta	IIT Delhi, India
Srinath Srinivasa	IIIT Bangalore, India
Sanjay Kumar Madriya	Missouri University of Science and Technology, USA
Masaru Kitsuregawa	University of Tokyo, Japan
Raj K. Bhatnagar	University of Cincinnati, USA
Vasudha Bhatnagar	University of Delhi, India
Mukesh Mohania	IBM Research, Australia
H. V. Jagadish	University of Michigan, USA
Ramesh Kumar Agrawal	Jawaharlal Nehru University, India
Divyakant Agrawal	University of California at Santa Barbara, USA
Arun Agarwal	University of Hyderabad, India
Subhash Bhalla	The University of Aizu, Japan
Jaideep Srivastava	University of Minnesota, USA

Program Committee Chairs

Ashish Sureka	ABB, India
Sharma Chakravarthy	The University of Texas at Arlington, USA

Workshop and Tutorial Committee

Anirban Mondal	Shiv Nadar University, India (Chair)
R. Uday Kiran	University of Tokyo, Japan

Panel Committee

Srinath Srinivasa	IIIT Bangalore, India (Chair)
Manish Singh	IIT Hyderabad, India

Publicity Committee

Punam Bedi	University of Delhi, India (Chair)
R. B. V. Subramanyam	NIT Warangal, India

Finance Committee

D. V. L. N. Somayajulu	NIT Warangal, India (Chair)
Naveen Kumar	University of Delhi, India

Publication Committee

Subhash Bhalla	The University of Aizu, Japan

Organizing Committee

Vikram Pudi	IIIT Hyderabad, India (Chair)
Suryakanth V. Gangashetty	IIIT Hyderabad, India
T. Raghunathan	IIITDM, Kurnool, India
M. Kumaraswamy	IIIT Hyderabad, India
Pratibha Rani	IIIT Hyderabad, India

Program Committee

Aastha Madaan	University of Southampton, UK
Akhil Kumar	Penn State University, USA
Alok Singh	University of Hyderabad, India
Anita Goel	University of Delhi, India
Asoke Talukder	Precision Genomics, India
Avishek Anand	L3S Research Center, Germany
Dhaval Patel	IBM, USA
Dhruba Bhattacharyya	Tezpur University, India
Durga Toshniwal	IIT, Roorkee, India
Enara C. Vijil	IBM Research, USA
Krishna Kummamuru	AI @ Accenture Operations, India
Lili Jiang	Umea University, Sweden
Lukas Pichl	International Christian University, Japan
Mandar Mutalikdesai	Siemens Technology and Services Pvt. Ltd., India
Manish Singh	IIT, Hyderabad, India

Mohammed Eunus Ali	Bangladesh University of Engineering and Technology, Bangladesh
Muhammad Abulaish	Jamia Millia Islamia, India
Naresh Manwani	IIIT Hyderabad, India
Niladri Chatterjee	IIT Delhi, India
R. K. Agrawal	Jawaharlal Nehru University, India
Rakesh R. Pimplikar	IBM Research, India
Ramakrishnan Raman	INCOSE, India
Ravi Madipadaga	Carl Zeiss, India
Renu Jain	UIET, CSJM University, India
Samiulla Shaikh	IIT Bombay, India
Santhanagopalan Rajagopalan	IIIT, Bangalore, India
Shady Elbassuoni	American University of Beirut, Lebanon
Shamik Sural	IIT, Kharagpur, India
Shubhashis Sengupta	Oracle Corporation, India
Soumyava Das	Teradata-Aster, USA
Srinath Srinivasa	IIIT, Bangalore, India
Sumit Bhatia	IBM Research, India
Swati Agarwal	IIIT, Delhi, India
Vadlamani Ravi	IDRBT Hyderabad, India
Vasudha Bhatnagar	University of Delhi, India
Vijay Srinivas Agneeswaran	Impetus, India
Yuanzhe Cai	University of Texas at Arlington, USA
Zeyar Aung	Masdar Institute of Science and Technology, UAE

Sponsoring Institutions

Special interest group on Big Data Analytics, Computer Society of India; Department of Computer Science and Engineering, NIT, Warrangal; International Institute of Information Technology, Bangalore; University of Aizu, Japan; Indraprastha Institute of Information Technology, Delhi; School of Computer and Information Sciences, University of Hyderabad, Hyderabad; Department of Computer Science and Engineering, Indian Institute of Technology, Delhi; Department of Computer Science, University of Delhi, India.

Contents

Big Data Analytics

Information and Knowledge Management

Mining of Massive Datasets

Computational Modeling

Data Mining and Analysis

Big Data Analytics

Smart Energy Management: A Computational Approach

Krithi Ramamritham[1](✉), Gopinath Karmakar[2], and Prashant Shenoy[3]

[1] Indian Institute of Technology Bombay, Powai, Mumbai 400076, India
krithi@cse.iitb.ac.in
[2] Bhabha Atomic Research Centre, Mumbai 400085, India
gkarma@barc.gov.in
[3] University of Massachusetts, Amherst, MA 01003, USA
shenoy@cs.umass.edu

Abstract. Among the practitioners in the energy management domain, there is enormous excitement about synthesizing and benefiting from numerous technologies, including real-time monitoring, net metering, demand response, distributed generation from intermittent sources such as solar and wind, active control of power flows, enhanced storage capabilities, and micro-grids. A common theme in today's solutions is the data-driven nature of the enabling technologies — to analyze requirements, use measurement/monitoring data to drive actuation/control, optimization, and resource management. The ability of modern sensing and IOT (Internet of Things) devices to inform us about the current state of the system and provide a timely and state-appropriate (rather than a broad, imprecise) response, backed up by analysis leads to novel solutions that are also practical and efficient.

Keywords: Smart energy · CPS · Smart grid · Smart home

1 Introduction

Energy management is all about making energy available whenever it is demanded by the consumers in order to maintain certain quality of life and sustain growth to meet the human development goals. Consumers want electrical energy to be available whenever it is required, be it for charging a mobile phone or their electric vehicle, running kitchen appliances or office copiers, or for indoor climate control using individual air-conditioners or huge chilling units.

Energy management requires monitoring, controlling, and optimizing the performance of all the elements of the electric grid in order to provide required energy of desired quality to the consumers. But, rapidly growing energy demand and the dependency on fossil sources to meet the gross as well as peak demand have raised concerns over poor quality of service (occurrences of black-outs, brown-outs and load shedding), depletion of resources and impact on the environment. Even major developed economies, such as the USA, have experienced

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 3–14, 2017.
https://doi.org/10.1007/978-3-319-72413-3_1

an increase in the number of major power outages over the past decade. The catastrophic blackout that India experienced in August 2012, which left more than 500 million people without electricity and basic amenities for several days, serves as a reminder of the urgency of acting on this challenge.

The widely acknowledged solution involves transitioning existing electrical grids to “smart grids”, a process that requires replacement of aging grid components, integration of renewable energy sources and energy buffering solutions, widespread deployment of sensors and actuators, and automating grid management using distributed Information and Communication Technology (ICT) systems [DOE 2012, Smart Grid Policy]. Concurrently, there has been an increasing focus on developing new technologies that will provide for a more sustainable future for our society. Since a significant portion of global energy use (and more specifically electricity use) continues to depend on traditional sources such as coal and natural gas, the use of novel ICT methods for the greening of energy has emerged as an important research area.

This is evident from two broad technology trends. *First*, there is a trend towards making the electric grid smarter, greener and more efficient. It is required that generation, transmission and consumption are to be kept in perfect balance in electricity supply system, whether a grid is smart or conventional. Any imbalance will cause disruption in the quality of electricity supply in the form of blackouts, brownouts or load shedding. What makes a grid smart is the way this balance between demand and generation is maintained, mainly due to the focus on automatic detection of the imbalance between transmission and distribution and taking *preventing* actions rather than taking just *protective* actions against system failures.

Today, there is an enormous excitement about synthesizing and benefiting from numerous technologies, including net metering, demand response, distributed generation from intermittent sources such as solar and wind, active control of power flows, enhanced storage capabilities, and micro-grids. Additionally, since building energy use represents a significant fraction of total energy expenditures, a *second* trend is the design of smart residential and office buildings. These have the ability to interface with the smart grid and regulate their energy footprint, reduce peak consumption, incorporate local renewable energy sources and participate in demand-response techniques.

Energy management systems are telling examples of cyber-physical systems (CPSs). To this end, we first examine the basic characteristics of CPSs. We then provide the necessary background and introduce the basic terms and concepts related to electrical energy. The need for a data and computation driven approach to energy management is discussed finally.

2 SMART Cyber Physical Systems

Cyber-Physical Systems (CPS) are made up of interacting networks of physical and computational components. These provide the basis for our critical infrastructure and for emerging and future smart services, and improve our quality of life in many areas.

Energy management systems are telling examples of CPSs. To this end, we examine the basic characteristics of cyber-physical systems.

Consider a digital temperature sensor (a smart device - more than a simple sensor), senses the temperature of the surrounding atmosphere through its sensor (e.g., thermistor or thermocouple), its ADC (analog-to-digital-converter) circuitry samples (with the help of a processor – usually a microcontroller) the sensor's analog voltage output with some specified frequency and produces a digital value, computes/processes the digital data to find the equivalent temperature value and finally responds either with a display of the temperature on the LCD or sends data over network. This device can also analyze the digital temperature data and compare it with a set value to generate an alarm if the temperature goes beyond the set limit. Here, we are essentially tracking the temperature and taking some simple actions.

For a more dramatic example, one with a lot of complexity, consider the systematic and fast evacuation of people during an emergency like fire, floods and bombs, a very important problem in modern society. With the increased threat of attacks by miscreants, this has become even more important. The problem has many dimensions. In case of fire and bombs in a building, the need is to quickly move people to the exits. In case of floods and water logging in a city/village due to disasters like Tsunami, people have to be moved to safe zones of the city/village. Since human life depends on the success of evacuation planning, a Smart Building Evacuation Planning System, which will help the building managers to evacuate people efficiently and systematically during an emergency such as scare, fires and terrorist attacks. In this case, sensors are deployed in the building to determine if a threat exists (such as a fire alarm). It will also use sensors to estimate the number of people present in the rooms and corridors of the building. We must also take into account the behavior of people during an emergency. Based on such information and the floor plans of the building, the system will suggest the routes that should be followed by during evacuations. Routes can be displayed using people's mobile phones, display boards and other notification mechanisms. The system is constantly executing the sense-analyze-respond cycle until it is known that it has no more work to do, i.e., nobody is known to be still in the building.

In this example, we are sensing the environment carefully to ensure that no person is present in the building, then depending on where people have been spotted we analyze possible solutions to choose the route for each person. We can extend such a building evacuation planning system to evacuate entire regions and cities. Of course, the scale of such an evacuation process will require highly efficient and scalable algorithms, along with low-cost and precise sensor technology.

Thus, a careful examination of how smart systems/devices/appliances work reveals that they have a certain pattern in their behaviour: a device meaningfully senses the parameter that informs the system about the current state of interest, analyzes the sensed value (often after some processing) to help in decision making and finally produces a timely response, which can be a decision or a value. Let us look at the three phases that embody a CPS.

2.1 Sense, Meaningfully

Smartness of any cyber physical system comes from accurate sensing of the environment and timely delivery of sensed data to analytics for further processing, analysis, control and further action. Sensor driven building management is driven by goals like reducing and optimizing power consumption, monitoring the health of the building appliances, maintaining quality of the atmosphere in the building and tracking occupants in various parts of the building (useful for building safety and emergency evacuation), to name a few. Different sensing and inferencing mechanisms are used to obtain the observations pertaining to different facets of the building.

The accuracy of sensed data and latency of communicating it to applications determines the quality of service (QoS) of a system. The accuracy of sensing may be affected by faulty or biased sensors while timely delivery may be affected by queuing and processing of increased data traffic in the communication infrastructure. Feeding inaccurate data for analytics or exceeding the latency bounds affects the performance of applications and thereby the reliability and responsiveness of the system. In practice, there is a tendency to over-provision sensors under the rationale that the more the data the more informed the decisions will be. Meaningful sensing relates to judicious sensing that ensures correct and timely decisions. Inaccuracy, unnecessary redundancy, or delays in sensing can make sensed values and hence decisions based on them meaningless.

A network of sensors is usually set up in the building by a BMS to obtain the information of interest. But installing these sensors in different parts of the building can (a) be tedious and expensive (b) cause inconvenience to the users (c) increase the payback period and, (d) affect the aesthetics of the building.

The fact that a sensor, suitable for observing a particular parameter, may in turn help to infer other parameters can be exploited to reduce the number of physical sensors deployed in a building. Similarly, inferences that are enabled by exploiting the structure of the building or the formal relationship between parameters, can lead to a better utilization of sensory resources.

2.2 Analyse

Analysis of the data sensed by the cyber-physical system has two major manifestations. One is based on archival or historical data. Another is on data pertaining to the prevailing situation. Given the online nature of decision making, i.e., we decide what is to be done in response to some real-world event, when the event happens, the response time is limited and hence we cannot always expect our choice of solutions to be optimal. Hence, often, to reduce the reaction or response time, the system analyzes the many possible solutions and also the system state in which a particular solution will be appropriate will be analysed and remembered by the system. When a real-time event occurs, the then state dictates the choice made.

2.3 Respond, Timely

Response is the action taken by the user or the CPS itself based on the analysis of the sensed data. Most “situations”, unless reacted to in time, will escalate. Hence the response of the system should be timely, many of the situations will have timeliness related requirements (e.g., deadlines). attached to them.

Because of the above characteristics of the sense-analyze-respond cycle, in many scenarios, special purpose hardware is designed for one or more of the three phases.

Sense **M**eaningfully, **A**nalyze and **R**espond **T**imely

A SMART approach results when we have a cyber physical system whose tasks will depend on the dynamics of the environment and whose responses are also situation dependent. Neither can be fully characterized statically (preventing just a table lookup to decide what to do at run time). Clearly, use of data from sensors to obtain situational awareness – state of both the environment as well as the system (resources) – is essential for meeting the challenges of such systems. Another crucial element is the synergy between the physical world and the ICT or cyber world.

3 The Smart Electric Grid

In this section we will take a brief look at the inadequacies of the traditional grid and show how the modern smart grid is being designed to overcome these.

3.1 The Grid of the Last 100 Years

Historically, the electric grid has served as a common interconnection network connecting power generators with consumers. At any instance of time, all the generated electricity is consumed entirely. In other words, balance is always maintained between the amount of generation and consumption.

A huge number of generators are interconnected through a network of power transmission lines so that electricity reaches the consumer with the highest level of availability via distribution lines. The grid can be viewed as a network of power transmission lines equivalent to links with generators as nodes. The distribution lines facilitate tapping of electricity at various points from the grid, and make power available to consumers, both industrial and domestic.

Ideally the electrical parameters, the voltage and frequency, of the grid remain unaffected even with changes in the electrical load connected to it. This is a necessity given that the loads are designed for particular voltage and frequency and their performance depends on the stability of these two electrical parameters. But, this does not mean that any amount of load can be connected or disconnected to the grid any time. The grid can accommodate fluctuations in load within its designed capacity.

3.2 Balancing Generation and Consumption

Generation, transmission and consumption are to be kept in perfect balance in the electrical grid – smart or conventional. Any imbalance will cause disruption in the quality of electricity supply in the form of blackouts, brownouts or load shedding. The good old grid system is no longer adequate to meet the present requirements – mainly in (i) offering support to diverse and large distributed generation, (ii) monitoring grid health by handling large volume of data in real-time so that faults can be prevented, rather than mitigated and (iii) facilitating consumers' participation in demand-response (D-R) control.

What makes a grid smart is the way this balance between demand and generation is maintained, mainly due to the focus on the automatic detection of transmission and distribution imbalance problem and taking *preventive* action rather than taking only *protective* action against system faults.

3.3 Peak Demand vs. Aggregate Demand

The demand for energy varies widely during the day. During certain hours in the morning and evening, demand is very high; such times are referred to as peak demand hours. Demand can go high during certain seasons like summer in tropical countries and winter in cold countries. Further, statistically, there can be sudden rise in demand in a grid with huge consumer base. In order to meet the peak demands, utilities must have provisions for additional generating capacities. Base load is managed by nuclear and large thermal plants – these cannot be brought in to meet sudden rise in demand, as it can take hours for them to start-up and get ready to be synchronized with the grid.

Mainly quick-responding oil/gas fired generating sources and hydroelectric plants are brought in to meet the peak demand. This is because they can be started within minutes and ramped up and down quickly to meet spikes in demand or sudden changes in the loads. While oil/gas turbine sources are inefficient and costly, the hydro generating sources have their own impact on the environment due to impact on the land as well as flora and fauna, aggravating flood situations.

Thus peak generators are economically inefficient and most of the utilities penalize the consumers, especially bulk consumers with higher tariff during peak hours. About 20% of the generating capacities exists in a power grid to meet the peak demand, which occurs only 5% of the time [1]. Therefore, flattening of peak demand is a need for improving economic efficiency.

3.4 Energy Storage and Renewable Energy Sources for the Smart Grid

Energy storage devices are becoming a vital part of the smart grid, especially in developing countries. Use of inverter batteries as a backup power source is common in residential buildings. UPS systems are often used for supporting critical loads and will be incomplete without batteries. This battery can also be used for storing power from the grid when the price of electricity is low. When demand for electricity goes beyond a threshold, charged batteries can be used to reduce the power drawn from the grid. This approach can help in saving energy costs for the consumer significantly. Charging and discharging decisions determine the saving opportunities. Also, renewable sources such as wind and solar photovoltaics require battery to store the excess energy generated in case the power cannot be pumped back to the grid.

Peak Demand is the highest amount of energy consumed during a time-slot over the billing period. Peak demand causes increased energy consumption at the same time, leading to significant imbalance, which may cause grid instability and failures within the grid. In order to discourage peak power demand, utilities charge higher consumption at higher rates. Hence, it is important to flatten the electricity demand. Peak demands are not only a problem from the consumer end but also for the utility to meet the demand. Normally utilities choose load shedding or power cuts if they are unable to meet the demand. Reducing electricity drawn from the grid during peak consumption periods by using batteries flatten the load profile that the grid discerns. In the electricity grid, electricity generation and consumption must be balanced at all times.

3.5 Conventional Grid vs. Smart Grid

From the above discussion, it is clear that the scope of smart grid is very wide and therefore a short yet complete definition of smart grid is not easy. This is evident from the following definitions from the Joint Report [2] of European Commission (EC), JRC and US-Department of Energy titled "Assessing Smart Grid Benefits and Impacts: EU and U.S. Initiatives, 2012",

According to EC [EC Task Force for Smart Grids, 2010a],

A Smart Grid is "an electricity network that can intelligently integrate the behaviour and actions of all users connected to it – generators, consumers and those that do both – in order to efficiently ensure sustainable, economic and secure electricity supply".

According to the U.S. Department of Energy:

A smart grid uses digital technology to improve reliability, security, and efficiency (both economic and energy) of the electrical system from large generation, through the delivery systems to electricity consumers and a growing number of distributed-generation and storage resources.

Table 1 summarizes the key differences between the conventional grid and a smart grid.

From what we have said thus far, we need a data-driven electric grid to make it smart. In addition, we also need all the consumers to be smart, that is, use the energy (made available to them) in a smart manner, and should design the generation decisions so as to be smart.

Many electric utilities are moving away from a flat pricing model to variable or peak usage-based pricing. In peak usage-based pricing, a utility monitors electricity usage over entire periods, such as every hour or every half hour, and bills customers, in part, based on the energy consumed in the peak period.

Table 1. Comparison between Conventional and Smart Grid

Topic	Conventional grid	Smart grid
Approach to power system faults	Detection and mitigation with a focus on protection of equipments	Focus is on prevention of fault by detecting emerging fault situations rather than responding only to the manifested faults
System monitoring and control	Monitoring of grid health is limited to small number of large power plants and no real-time information for adaptive protection	WAMS (Wide Area Measurement System) enabled by ICT to convey real-time information for improved monitoring and almost instantaneous stability of supply and demand on the grid
Integration of renewable generations	Not equipped to support Distributed Energy Resources (DER)	Supports diverse and distributed generations with a focus on renewable resources
Power Quality (PQ)	PQ mostly neglected with focus on minimizing outages	Ability to identify and resolve PQ issues like voltage fluctuations, interruptions, waveform distortions prior to manifestation
Consumers participation	Uninformed Consumers have no role to play in the power system management	Two way communication and active involvement facilitating deeper Demand-Response penetration

So, a decrease in total and/or peak usage results in a more than linear reduction in the monthly electricity bill.

4 Solutions for Energy Based on Computational Thinking

We believe that the so-called "computational thinking" can lead to smart energy management solutions. Specifically, data driven approaches and use of Artificial Intelligence (AI) techniques such as inference and learning are essential for smart energy management, to increase the energy efficiency.

Efficient use of energy is an age-old goal. But its importance has become even more apparent with the increased emphasis on human development and the increased use and thirst for more energy that it engenders. The focus on addressing the energy concerns through the use of information and communication technologies (ICT) has two implications: (i) Harness today's processing and communication tools to improve the efficiency and responsiveness of existing energy management systems. (ii) Use the ability of modern sensing and IOT (Internet of Things) devices to inform us about the current state of the system and provide a timely and state-appropriate (rather than a broad, imprecise) response, backed up by analysis. This goal will drive us to make use of recent research trends in data driven methods for improving energy-efficiency of buildings, campuses, and cities.

There is enormous excitement about synthesizing and benefiting from numerous technologies, including real-time monitoring, net metering, demand response, distributed generation from intermittent sources such as solar and wind, active control of power flows, enhanced storage capabilities, and micro-grids. Additionally, since building energy use represents a significant fraction of total energy expenditures, a second trend is the design of smart residential and office buildings. These have the ability to interface with the smart grid and regulate their energy footprint, reduce peak consumption, incorporate local renewable energy sources and participate in demand-response techniques.

4.1 Data-Driven Approaches

A common theme in today's solutions is the data-driven nature of the enabling technologies — to analyze requirements, use measurement/monitoring data to drive actuation/control, optimization, and resource management.

A smart building needs to monitor the *states* of the building in terms of (i) power consumption, (ii) possible wastage of power, occupancy, (iii) thermal comfort level etc. This requires a wide range of sensors so that the building management system can monitor and carry out its tasks of reducing and optimizing power consumption, monitoring the health of the building appliances, maintaining quality of the atmosphere in the building and tracking occupants in various parts of the building (useful for purposes such as building safety and emergency evacuation) etc.

4.2 Taming Big Data with Analytics

Since there is a need for many different types of sensed data it is not feasible to deploy sensors "everywhere". Doing so can greatly increase the cost and payback period, while also impacting building aesthetics and causing user inconvenience. Furthermore, this will cause a great deal of redundancy in the sensing infrastructure.

First, there is redundancy due to the inherent structure of the sensing problem. An example, consider Fig. 1 where sensors are placed at every node of a tree, where the leaf nodes represent individual power outlets, while the parent node represents the circuit breaker that supplies power to these outlets. We observe that the sensor on at least one of the nodes is redundant. Thus, if we measure power usage at each outlet, the total usage of the circuit can be obtained as the sum of the power used by the individual outlets. Similarly the power usage of an outlet can be computed from that of the breaker and the other two outlets.

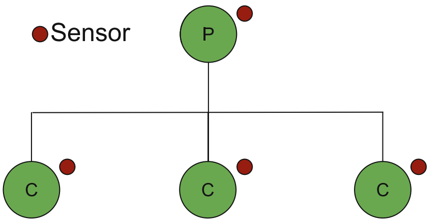


Fig. 1. Redundant sensing

In some cases, there may be similarities in the power usage of the leaf nodes. For instance, the same type of load may be plugged into all three, or the leaf node may represent the power usage of an office room and all office rooms may be similarly equipped. In this case, *sampling* one or a small number of leaf nodes may be sufficient and the power usage of the other leaf nodes can be estimated from these samples.

Second, there may be redundancy from different types of sensors. For example, one type of sensor deployed in an area may reveal other types of information, typically obtained using sensors dedicated for that purpose. measured by other sensors. As an example, consider motion-activated lighting. Since the motion sensor turns lights on or off based on motion (occupancy), there is no need to separately monitor the power usage of the lights, since it can be directly computed based on whether there was motion in the room. The reverse is also true – monitoring power usage in an area can also reveal occupancy, since more power events (e.g., manually turning lights on or off) indicates the presence of users in that area.

Third, a building may include soft sensors that are part of the IT infrastructure that may reveal certain types of information, making hard sensors redundant. For example, WiFi activity at an access point or active network traffic from a desktop reveals the presence of humans in the vicinity, and can be used as soft sensors for tracking occupancy in that part of the building.

Fourth, certain types of data can be inferred using more sophisticated machine learning and inference algorithms that take inputs from other hard (physical) or soft sensors. Load disaggregation algorithms that infer the power usage of individual loads from aggregate measurements of total power usage is an example of type of redundancy.

Fifth, depending on the application goals, it may be possible to exploit redundancy in how many sensors need to be deployed. For example, if an application needs to monitor whether a home is occupied, one approach is to deploy motion sensors in every room. Another approach is to deploy door sensors that track how many people enter and leave the home, maintaining a count of how many people are inside. If the application only needs to know whether the building is occupied, the door sensors are adequate for this task and room-specific motion sensors can be eliminated. However if the application also needs to know spatial occupancy details of which rooms are occupied, the door sensor is inadequate and each room needs to be instrumented with motion sensor. Thus, the same problem of occupancy monitoring may require different levels of instrumentation depending on the goals of the higher level application.

Thus, it is possible to exploit these redundancies to vastly reduce the number and types of sensors and actuators that need to be deployed without sacrificing quality and resolution of sensor data while meeting application-specific goals. To answer our previous question of what sensors to deploy in a building and where, we must consider the following issues:

- Different parts of a building have similarities in their consumption profile.
- Redundancy in hierarchical nature of the building sensing infrastructure.
- Observations made in the context of one facet to infer another facet of interest.
- Soft sensors can minimizing the need for physical sensing.
- Goals of the specific energy management tasks (i.e., application goals).

The SMART model has been applied in the context of buildings to address areas that demand attention to make a building smart. These are (i) smart sensing, (ii) modeling electrical loads, analysing their pattern of operation and power consumption and (iii) offering suitable control action to achieve the objective of reduction in energy consumption, flattening of peak demand and providing thermal comfort to the consumers. Smart sensing requires the gamut of sensors to obtain the building state –with optimal deployment of sensor resources related to occupancy, power consumption, thermal conditioning, status' of appliances etc., necessary to meet the objectives of smart buildings. Classification, modeling and analysis of electrical loads in buildings are done with a view to facilitating higher level energy optimizations such as flattening of peak demand and reduction in consumption.

Then there is the focus on achieving thermal comfort for users of buildings. Heating and cooling is the dominant contributor to the energy consumption of buildings. Reducing the energy consumed due to heating and cooling while ensuring thermal comfort for building occupants is therefore an key challenge. We have to take into consideration various considerations in providing thermal comfort, factors influencing thermal comfort, the stages involved in providing thermal comfort given the lifetime of a building, undesirable phenomena requiring pro-active and reactive interventions along with a description of the many possible interventions.

Computational approaches for SMART energy management of solar and wind-based renewables. Storage can be exploited to address the challenges of intermittent generation. Electric vehicles (EV) and their impact on the electric grid and techniques for smart EV charging. Forms another area of research.

It is required to develop techniques for incentivizing users to become energy efficient, techniques for exploiting electricity pricing for smart energy management, energy trading techniques, techniques for campus-scale energy management, big data approaches for city-wide energy management, and security and privacy issues.

5 Conclusions

The Broad Goals of Smart Energy Management include:

- Providing power to consumers, ensuring quality - with higher availability at a lower cost.
- Enabling energy conservation – to decelerate the fast depletion of nonrenewable resources.
- Reducing the dependence on unsustainable energy sources – by avoiding unnecessary consumption.
- Increasing the use of sustainable energy sources – by exploiting renewables and finding ways to store excess energy from the sun or wind during periods of low consumption.
- Achieve peak shaving - by staggering loads or by scheduling appliances at the right times.
- Ensuring user convenience or comfort by automating tasks, providing timely feedback, or ensuring a comfortable environment.
- Incentivize users to become more energy efficient and adopt a more sustainable lifestyle.

The field of smart energy management has seen fairly active research in the last few years, but is still in a state of flux and many interesting problems remain. It is mature enough that some products have hit the market, but not stable enough to deter fresh startups from entering the arena. Further research will help accelerate the cross-fertilization of ideas from multiple disciplines.

Our goal was to demystify the ICT based solutions to people who study this problem from an electrical engineering or energy science and engineering perspective and for the IT and CS researchers and practitioners to be able to approach the energy issues with some comfort.

References

1. Farhangi, H.: The path of the smart grid. IEEE Power Energ. Mag. **8**(1), 18–28 (2010)
2. Giordano, V., Bossart, S.: Assessing smart grid benefits and impacts: EU and U.S. initiatives joint report EC JRC - US DOE. ISBN 978-92-79-26477-1 (pdf), ISBN 978-92-79-26478-8 (print), European Commission - JRC and US Department of Electronics, Technical Report (2012)

Big Data Analytics Enabled Smart Financial Services: Opportunities and Challenges

Vadlamani Ravi[1(✉)] and Sk Kamaruddin[1,2]

[1] Centre of Excellence in Analytics, Institute for Development and Research
in Banking Technology, Castle Hills Road No. 1, Masab Tank,
Hyderabad 500057, India
padmarav@gmail.com, skkamaruddin@gmail.com
[2] SCIS, University of Hyderabad, Hyderabad 500046, India

Abstract. Of late, the financial services industry is fast moving away from the traditional paradigm to the sophisticated digital way of dealing and the customer. Both the facets of the financial service industry, viz., the financial service provider and the customer are going through a digital evolution. In particular, banking industry has evolved from just journal and ledger entry paradigm to data and analytics driven banking operations, which subsumes online as well as offline customer behavior. This paper discusses various scenarios in baking, finance services and insurance (BFSI) areas, where big data analytics is turning out to be paramount. The paper also highlights the potential benefits, of the new-age technologies viz., Internet of Things (IoT), Blockchain, Chatbots and robotics.

Keywords: Big data analytics · Digital banking
Financial services · IoT · Chatbot · Insurance · Hadoop · Spark

1 Introduction

Financial transactions evolved over time from the ancient barter system to today's state-of-the-art e-commerce system. With the rapid advancement of human civilization and the associated stellar technological achievements, the finance services industry (also known as banking, financial services and insurance -BFSI- industry in India) has thrived significantly. Earlier to the digital era, all the transactions and the business intelligence thereof had excessive human involvement. The digital world, while making the transactions clear and transparent, generated large amount of digital data. These digital footprints have become amenable for rigorous analysis using the new field called analytics in order to make clear and right business decisions. Over a period, with the ever-growing young customers, their increasing needs & desires and globalization, financial services industries produced humongous amount of varieties of data at a break-neck speed leading to a new generation of data analytics paradigm called Big Data Analytics (BDA).

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 15–39, 2017.
https://doi.org/10.1007/978-3-319-72413-3_2

1.1 Introduction to Big Data

Data-driven technologies and decision making is often called fourth paradigm of science, the theoretical, experimental and computational paradigms being the other three. Over the past two decades, many science & engineering disciplines, medicine, business, economics produced vast amounts of data in various forms thanks to the proliferation of sophisticated instruments, cheap hardware and novel business processes. This trend is exacerbated by the rampant use of social media via Web 2.0. Analyzing this huge data and providing a better consumer experience with better data management have led to the genesis of Big Data Analytics (BDA). These monetary transactions in banking, finance, service and insurance sectors generate a huge volume of data in less span of time from different digital devices involving different types of data formats. The big data is best characterized by 5Vs. They are Volume, Velocity, Variety, Veracity, and Value. *Volume* refers to the vast amount of data generated every second. The huge amount of transactions result into a humungous datasets, which are difficult to store and analyze with the help of conventional storage and computational technology. The big data paradigm allows us to store these datasets and facilitates their computation by employing the distributed storage and parallel computational framework. *Velocity* dimension refers to the speed at which new data is generated and the speed at which data moves around. *Variety* refers to the different types of data that is generated. The big data technology deals with different types of data including text data viz. SMS, social media conversation, feedbacks in websites, tweets in twitter, image data viz. photos posted on Facebook, Instagram, and images gathered from satellites, sensor or sensor embedded device data, and video or voice recordings; and unite them with traditional structured data. Thus, this dimension subsumes unstructured, semi-structured and structured data from various sources. *Veracity* dimension presents the degree of reliability of the data. The presence of structured, unstructured, and semi-structured data in big data renders its quality and correctness less controllable. Therefore, this dimension concerns the uncertainty associated with the data. Finally, *Value* dimension refers to the business value that can be extracted from the data [1]. Very often, the noise component in the data collected is disproportionately more compared to the useful data present thereof. Thus, these five dimensions succinctly capture the entire characteristics of big data.

1.2 Introduction to Apache Hadoop

As the data generated by the Banking, Finance sectors and Insurance is of large scale, with different data formats, which are not possible to handle with the traditional relational database. Here comes the open source Apache Hadoop framework, which can store a large volume of data with the help of distributed storage file system and process or analyze them in a distributed manner with the help of parallel MapReduce computational architecture.

The base Apache Hadoop framework comprises the following modules: (i) Hadoop Common, (ii) Hadoop Distributed File System (HDFS), (iii) Hadoop Yarn, and (iv) Hadoop MapReduce. The Hadoop common module contains libraries and utilities required by other Hadoop modules. The HDFS is the distributed file system, which stores the data distributed over several commodity machines present in the cluster. YARN is the platform that allows the management of computational resources pre-sent in the cluster and scheduling the users' application. MapReduce provides the computational framework implementing distributed processing for large-scale data [2].

1.3 Introduction to Apache Spark

Apache Spark is a fast, distributed computing technology. It employs horizontal clustering for fast and efficient computation. Apache Spark provides its computational framework on top of Hadoop MapReduce (MR) model, and it employs MR model for an extended computational framework subsuming interactive database queries and online processing through streaming. The most striking attribute of Spark is in-memory computation, which reduces the read/write latency of intermediate data during processing.

Spark can handle different workloads such as batch program, iterative codes, interactive database queries and streaming data. The Spark is faster than Hadoop distributed processing, and it is attributed to the reduced amount of read/write tasks to the hard disk. It stores the intermediate value of the variables in memory during the execution [3].

The Spark core is the computational framework that provides in-memory computation. Spark core engine supports APIs in Scala, Java, Python or R. Spark also supports 'Map' and 'reduce' operations. In addition to it, Spark supports Machine learning (MLlib), SQL queries (Spark SQL), Streaming data for online processing, and Graph algorithms (GraphX). The shared memory facility of sharing the same data among multiple applications is achieved through Tachyon now called, Alluxio. The cluster management in Apache Spark is performed in three different ways viz. Standalone, Hadoop Yarn, and Mesos. Spark can access the data from local file system or any distributed file system like Hadoop Distributed File System (HDFS).

2 Current Works and Recent Trends

The current scenario of the financial world reflects the utility of data and analytics carried out on them for the insights. The analytics, and in a more formal way the big data analytics has paved a path to present another dimension to the business. The big data analytics can be implemented to address several problems the financial industries are facing during their operation.

2.1 Digital Banking

The digital world has made a revolution in the banking industry. Initially starting with core banking now the banking industry has moved to multichannel banking industry with different types of devices. In general, a bank has external and internal aspects. The external facet refers to the customers (both retail and corporate) of the bank, the regulator and other competing banks and partners; the internal facet includes treasury, back-office operations, and HR department. Considering these two aspects, a digital bank has eight predominant dimensions that form the complete umbrella for a digital banking system (Fig. 1).

The external aspect subsumes regulatory and operational aspects of banking. In turn, operational aspect subsumes the dimension of customer/sales and services. These are the first two circles in Fig. 1.

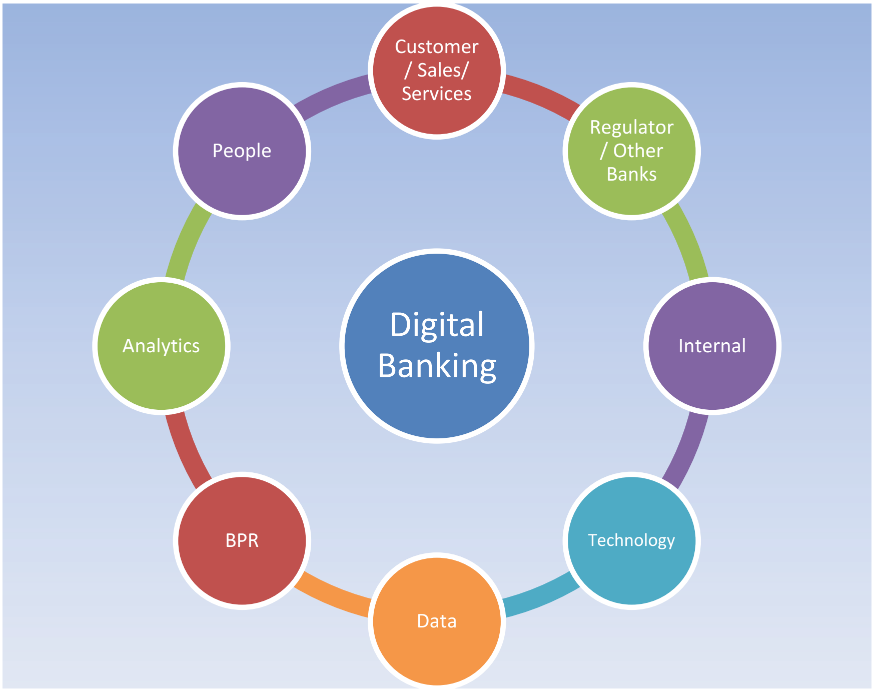


Fig. 1. Dimensions of a digital bank

Customer/Sales/Services. It is the predominant pillar of any digital banking framework. For providing the financial services and sales to the customer, a digital bank is supposed to implement holistic CRM (subsuming operational, analytical and collaborative CRM). Customer-centric business models are established based on an exhaustive knowledge of the customer. Such a model demands tactical planning on:

- Establishing an omnichannel integrated platform – to facilitate seamless and consistent user experience across all the channels, i.e., internet, banking app over smart phone and social networking services.
- Developing the competence to collect, incorporate and analyze numerous sources of internal and external data – to comprehend the customer for a personalized solution.

- Comprehending and interpreting relevance and timeliness for the customer – to customize processes from the perspective of the customer. Analytical and prescriptive CRM play a predominant role in the digital banking journey.

Regulatory/Other Banks. It comprises a seamless communication at several business levels and fraud-related reports to the regulatory board, e.g., RBI. It also involves seamless communication between various commercial banks to have smooth banking operations.

Internal. It comprises applications for analytics with measurement and management of different types of risks a bank can face during its day-to-day operation. These includes credit risk, market risk, operational risk at the highest level and many other risks at a lower level.

Technology. This aspect involves core banking, internet and mobile banking, e-wallets, m-wallets, Omni-channel data warehouse, Data Lake and service oriented architecture.

Data. The data dimension deals with the quality of the data that is present within the bank. This deals with data governance and data quality management.

Business Process Reengineering (BPR). This dimension deals with redesigning and reengineering the existing business processes in order to be more customer friendly and profitable.

Analytics. This dimension drives the bank towards profitability while making it customer friendly simultaneously. This dimension involves three types of analytics viz. (i) descriptive, (ii) predictive, and (iii) prescriptive. The descriptive analytics presents the information content available in the historical and current data by answering complex, high dimensional queries in pictorial forms viz. bar chart, line chart, pie chart, histograms, heat chart, box plots, etc. The predictive analytics solves various business problems faced by a bank based on historical data and utilizing several applied statistical and machine learning models. It helps the organization to proceed in the right direction and take right step at right time. The prescriptive analytics consists of applying optimization techniques to recommend future course of action based on the predictions made in predictive analytics stage.

People. This dimension deals with the manpower involved in the banking industry to lead to a digital banking era. For that, the bank requires the recruitment of qualified and trained specialist manpower for transforming the bank into a revenue generating organization as well as a customer friendly one.

The digital banking generates a large volume of digital data from which meaningful predictions and insights can be obtained using predictive and prescriptive analytics. Analytics in digital banking manifests in six distinct ways, viz., customer analytics, fraud analytics, risk analytics, operational analytics, security analytics, and HR analytics (Fig. 2).

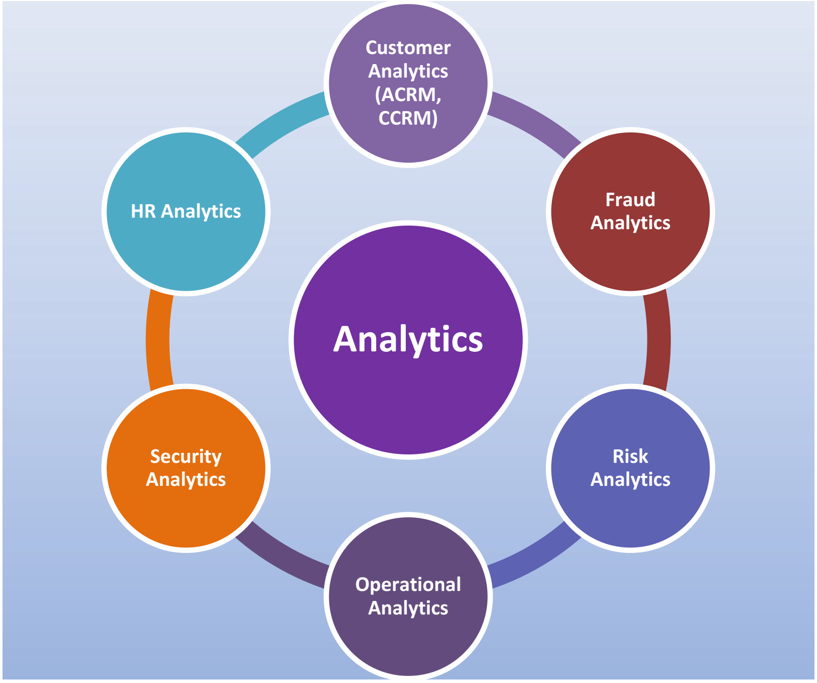


Fig. 2. Analytics through digital banking

Customer Analytics. It is also known as analytical CRM (ACRM). It produces the 360° view of a customer through dashboard tool. The dashboard presents the different derivable insights for the customer utilizing Marketing Analytics, Channel Analytics, Social Media Analytics, Collections or Recovery Analytics, Collaborative CRM, etc. All these flavours of analytics provide instant decision-making ability on the part of bank agent for an operational task. The research community has contributed in the Customer Analytics domain.

Jayakrishna and Ravi [4] surveyed the evolutionary computing methods applied to analytical CRM. They found that the evolutionary computing has been applied to single as well as multi-objective problems in analytical CRM. The review work presents the maximum work has been carried out in market basket analysis, followed by credit scoring and fraud detection. According to the review work, the most explored technique is Genetic Algorithms.

Erevelles et al. [5] proposed consumer analytics in big data paradigm for gaining the sustainable competitive advantage for the business over the competitors. The customer insights will result in value creation concerning product, price, place, and promotion.

There are many research works carried out in customer analytics, which involve text analytics analyzing the textual data generated through multiple channels. These involve tweets, Facebook posts, WhatsApp messages, feedback at call centers and e-mails posted by the customers related to financial products and services.

Fraud Analytics. It covers cyber (online) and offline frauds perpetrated in the cyber interface of banks viz., ATM, internet banking, mobile banking and credit/debit cards or in the physical branches of banks. Thus, fraud can be perpetrated by employees from within the bank branch in collusion with customers/third parties or outside the bank's purview. In general, the sophisticated cyber frauds involve a group of fraudsters from various geographical regions linked together through telephones, social networks, e-mails etc. The social network analytics, social media analytics, and text analytics come in handy to detect and jeopardize the fraudsters' plans.

Risk Analytics. It comprises all types of quantifiable risks viz. credit risk, market risk, and operational risks. All these types of risks are quantifiable via analytical models and thanks to the vast amount of data can be predicted using the data-driven methods. These risks are modeled and analyzed for different operations in banking such as granting a personal loan to a customer by the computation of credit risk involved in granting the loan.

Kshetri [6] presented how big data is utilized in appraising, evaluating and filtering the creditworthiness of customers for loan and minimizing the transaction costs. The author also presented how the different features of big data such as volume, velocity, variety, etc. are related to the assessment of the creditworthiness of low-income families and micro-enterprises present in China.

Sun et al. [7] analyzed big financial data utilizing wavelet-based methods for predicting the change in the price of equities. They proposed a new wavelet-based method called GOWDA, which forecasts the volatility of equities in an efficient manner.

Operational Analytics. It covers all operational problems of a bank viz., ATM cash replenishment strategy, ATM/branch location problem, balanced scorecard based assessment of a bank's growth, assessment of a bank's performance w.r.t profitability, solvency, productivity, liquidity, etc., modeling gridlock scenario in inter-bank payments, etc.

The financial operations generated huge amounts of digital data where a major portion is from the payment channels used for transactions. There is a huge change in the preference of usage of payment channel from 2007 to 2017 (Fig. 3). Here it is clear that the new-generation, young customers prefer not to go to the branches or the ATMs.

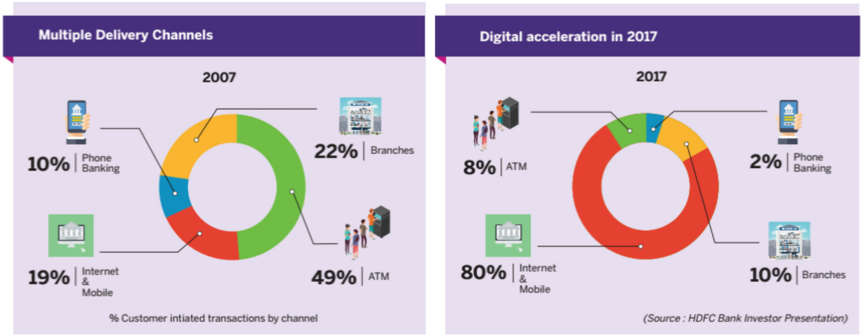


Fig. 3. Trend of payment channel usage [8]

Instead, they prefer to perform their transactions at their convenience with their own time. This change of preference has made a great shift in Internet and Mobile banking penetration from a mere 19% in 2007 to 80% in 2017 [8]. Therefore, the banking industry can leverage the huge amounts of digital data generated thereof to discover the valuable insights. These insights, in turn, will be financially beneficial to the bank and eventually may improve customer experience.

Akter et al. [9] proposed a model for Big Data Analytics Capability (BDAC) for enhancing the firm performance. The BDAC subsumes BDA Management Capability (BDAMAC), BDA Technology Capability (BDATEC), and BDA Talent Capability (BDATLC). BDAMAC is responsible for right business decisions by employing proper management framework. BDATEC refers to the capability of the model to connect multi-source data, to work on cross-platform to develop, deploy and support a firm's resources. BDATLC refers to the ability of the analytics professionals to carry out the tasks in the big data environment. This model ensures the enhancement in firm performance.

Social Network Analysis (SNA) in Banking for Fraud Detection. The Social Network Analysis (SNA) is posing itself as a vital tool for financial fraud detection. The different fraud statistics show that the credit card fraud is increasing with the card-not-present (CNP) type of fraud. This type of fraud involves a group of fraudsters working in collaboration, where the SNA method is establishing an impressive fraud identification technique. The social network is a network of entities all connected in a particular way. The entities can be credit cards, companies, merchants, fraudsters, or others. It can include transactional data, such as online transactions and banking data, social media data, call behavior data, IP address information, geospatial data, etc. This data is often stored in unstructured formats in environments like social media, telecom registries, payment gateways or bank servers.

Storing and retrieving interconnected information in a native 'network graph' format can deliver interactive network visualizations to discover hidden structures, locate clusters and patterns, identify links in transaction chains, and apply specialized algorithms to identify suspicious patterns.

Traditional methods for detecting fraud involve analysis of the risk score of the retail of enterprise customer for the creditworthiness. Banks use applications that would set some decision rules into a model, and the model would scrutinize to determine whether to approve or disapprove a transaction based on these rules. The SNA method tries to predict what constitutes fraudulent behavior. It provides a method for analysis of the relationship among the fraudsters across multiple channels through the link analysis in a social network graph. In simpler words, it is the representation of distinct subjects (nodes) and their association (edges) through a graph [10].

SNA can furnish effective insights from large-scale datasets along different dimensions viz. network, spatial and time. It is based on the analysis of interconnectedness of the subjects present in the social network graph [11].

For the financial service industry, SNA has to play a key role in recognizing fraud. This is because frauds are to a large extent being perpetrated by organized networks. In payments and banking sector, the social networks comprise an account number, card number, customer name, email address, phone number and so on. Relationships (edges in the graph) between these various nodes can be scrutinized to diagnose patterns that could reveal fraudulent behavior (Fig. 4).

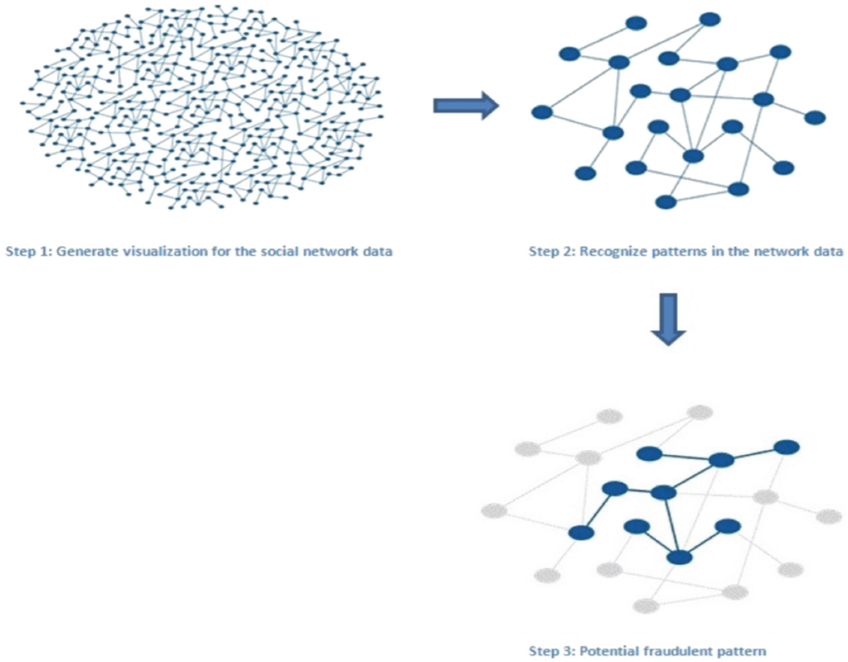

Step 1: Generate visualization for the social network data
Step 2: Recognize patterns in the network data
Step 3: Potential fraudulent pattern

Fig. 4. Detection of fraudulent patterns through SNA [12]

Chau and Faloutsos [13] had made a case study for discovering financial frauds employing a method utilizing the *belief propagation* algorithm [14].

Saidi et al. presented a review paper reviewing the different approaches for analyzing cyber terrorist communities. The relational analysis and positional analysis can reveal the terrorist network by analyzing the social network graph. The relational analysis studies the interactions between network members, between nodes within a graph. The positional analysis focuses on how two members within a network are similar taking into account their connections to other members to discover a social structure in a network. The different SNA measures like Size, Density, Degree of connection, Centrality like the Degree centrality, Betweenness centrality, Closeness centrality, Eigenvector centrality are used for terrorist network analysis [15].

Text mining also plays a major role in fraud detection and prevention. The text can be analyzed from the online feedbacks, social sites, etc. Shravankumar and Ravi [16] reviewed the applications of text mining for cybersecurity in the financial domain by covering Malware detection, Phishing detection, Spam detection, Fraud and Intrusion detection. The review finds that the most explored techniques are Decision Tree, Support Vector Machine, Naïve Bayes, k-nearest neighborhood for cybersecurity detection.

West and Bhattacharya [17] reviewed the different types of financial frauds. The different types of financial frauds reviewed were credit card fraud, securities and commodities fraud, financial statement fraud, insurance fraud, mortgage fraud, and money laundering. In literature, both the statistical and intelligent methods were proposed for financial fraud detection. The reviewed articles present Credit card fraud has been analyzed with Support Vector Machine (SVM), Decision Tree (DT), Self-organizing Map (SOM), Fuzzy logic, Artificial Immune System (AIS); securities and commodities fraud has been analyzed with Bayesian Belief Network (BBN), Process Mining; insurance fraud has been analyzed with Logistic model; and financial statement fraud has been analyzed with Neural Networks (NN), DT, BBN, SVM, GA.

Security Analytics. It incorporates vulnerability analysis, advanced persistent threat prediction, intrusion detection, data exfiltration detection, anomaly detection, phishing detection, spam detection, malware detection, DDoS detection, SQL injection attack detection etc. The Fig. 5 succinctly captures the multi-disciplinary nature of cyber security, wherein multi-pronged strategy is proposed to be adopted to predict and prevent cyber security incidents. Of the six different, albeit slightly overlapping dimensions, data analytics play a significant role because they heavily depend purely on the various types of data generated and found around security incidents.

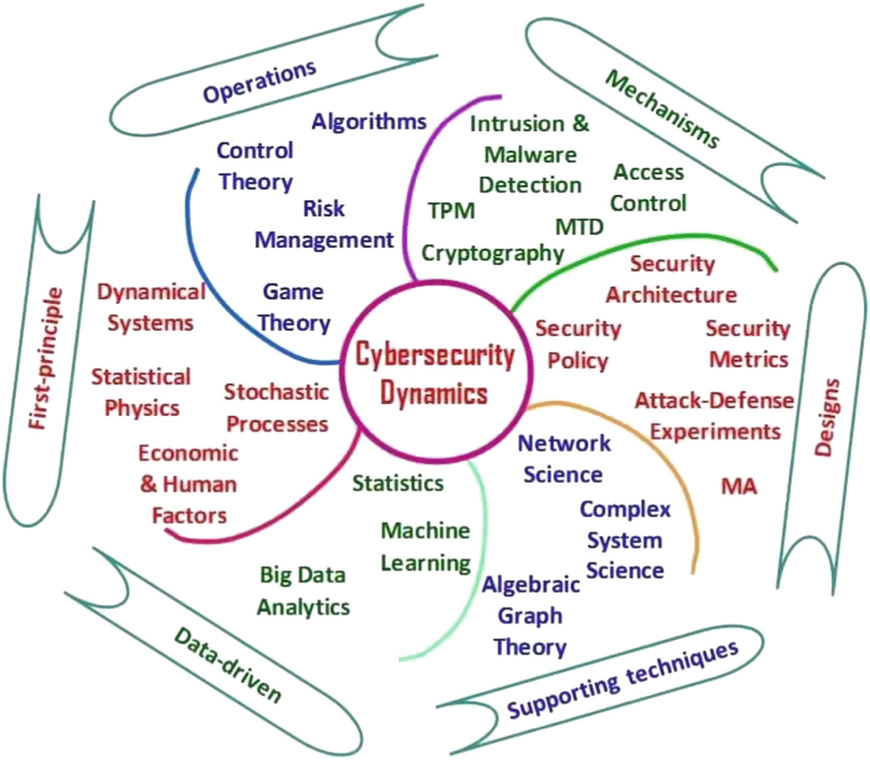


Fig. 5. Cyber security dynamics [20]

Digital Forensics in Banking and Financial Services. It is also known as cyber forensics is a branch of forensic science comprising the collection and analysis of the materials found in digital devices, basically related to online or offline computer crime [18].

In simpler words it can be also defined as the science of identifying, preserving, recovering, analyzing and presenting facts about digital evidence found on computers or digital storage media devices [19].

The process of digital forensics can be broken down into five parts. They are as follows.

Identify. The digital forensic process starts with identification. On every crime scene, the evaluation of environment is started with identification of the data, which is also known as *artifacts*. The different devices and data transfer among them leaves a different type of artifacts, which has to be identified as the first part of the process.

Preserve. It is the crucial part of the digital forensics, which ensures the integrity of the evidence. As without integrity an evidence loses its value. Hence, utmost care should be taken to ensure the artifacts are not altered and preserved in its original state.

Recover. The recovery process deals with recovering (i) intentionally deleted files from the system, (ii) data from password-protected files, (iii) even from damaged or corrupted files or devices.

Analyze. This phase starts only when artifacts are identified and recovered. Analysis is the main part of the investigation. This analysis involves recovery of artifacts from the linked or synced devices viz., search history, download or upload history, chat history, etc.

Present. Finally, when the analysis is over, the findings are presented in the form of a case report. This part presents the facts precisely and concisely.

HR Analytics. It provides insights on the possible attrition by performing social media analytics. It can also help the HR in recruiting the right person for the right job at right remuneration.

The digital payment channels are making a gradual paradigm shift from personal banking to digital banking. The customers now prefer to digital banking due to the busy life schedule, less convenience to move to the bank premises and the high availability of smart phones. Some banks saw opportunity here and made a complete banking solution through digital banking and in particular through mobile banking. Table 1 presents products offered by some of the mobile-only banks. More details can be found at [21].

There are several Apps, which incorporate the Artificial Intelligence (AI) to ease the human day-to-day life activities. The apps can also be utilized for decision making in BI. These are also known as a virtual personal assistant. They are listed in Table 2.

Apart from the above-listed apps, there are some more apps which use AI to achieve their objectives. Such apps are viz. Braina, PAN, Wipro Holmes, Kinect, Wolfram Alpha.

Table 1. Mobile-only banks with products or services

Region	Name of the bank	Description
UK	Atom Bank	• It provides usage of biometrics instead of passwords in the mobile app for banking • It offers convenient banking methods to manage money
	Monese	• It offers the customers regardless of their citizenship, to open an account using the mobile banking app in minutes • It performs targeted marketing by allowing and helping the immigrants to open a UK bank account, e.g., a current account and also the issue of a Visa debit card in 3 min with a snapshot of their passport and a selfie
	Osper	• It provides debit card and mobile banking service to manage one's finance • It offers accounts with separate logins for minors and their parents
	Mondo	• It offers the banking app in iPhone and a prepaid debit card for managing the account • The banking app provides real-time feedback regarding customer's spending
	Starling	• The bank provides a high-quality current account service
US	Simple	• It combines user's experience and behavioral economics with technology, to result in proper insights to allow the customers spend intelligently • The account offered by the bank has all the variety of tools that a customer may need to manage his finance, which can be accessed through internet, and smartphones using iOS or Android OS
	Moven	• Its app and debit card provide insights from the transaction history on a real-time basis to the customers to make proper decision to manage their finances
	BankMobile	• It is the first bank of its kind that provides free savings accounts and a line of credit. A customer can avail access to more than 55,000 surcharge-free ATMs
	GoBank	• It is the first bank that offers an account to be opened and managed from only an app installed on a smartphone • It facilitates to check balances on the account, transfer funds or to observe transaction history
India	digibank by DBS	• It is founded in the year 2016. It permits to deposit, withdraw and transmit money. It allows a customer to set financial targets and outline an action plan to achieve them • It is the first bank in India that allows opening of an account without paperwork. It requires Aadhaar number and biometrics for authentication • It allows the customers to experience the banking facilities with an e-wallet and later transform to a full-fledged bank account at one's convenience

Table 2. Some popular virtual personal assistants

Virtual personal assistant	Features and functionality
Siri	It is developed by Apple for iPhone users. It is a voice-activated virtual assistant. It returns customized answers after learning from a user's language. As of 2016, Siri is available in 20 languages. Performing online search, making calls by the voice commands with the name or number, tweeting with voice commands, setting reminders and events in the calendar, setting the alarm are a few things that Siri can perform nicely. It can also launch applications from the device, find and readout emails with the audio. It works only on iOS operating system
Google Now	It is an intelligent personal digital assistant (PDA) by Google, which uses NLP, enabled interface. It can perform actions based on voice commands or typed commands. It runs on devices with different architecture and OS. It performs quite similar to Siri such as reading out texts and emails, performing an online search, making an online reservation on flight or train, updating the real-time score from stock and sports, etc. Google Now fetches desired results by utilizing a lot of personal information, which brings privacy concerns into the picture
Cortana	It is the PDA developed by Microsoft. It can recognize both voiced and typed commands. It facilitates NLP based search, detects songs in the playlist of the device and plays it. In addition to, it can open websites in the browser, send emails, create alarms and chat with. It cannot launch applications from the device like Siri and change settings of any device or any application. It has most powerful AI in the form of Adam's deep learning. Cortana can also analyze an image, e.g., it can inform the user about the amount of calories present in the food from its image
Alexa	Alexa is the voice service created by Amazon for Amazon Echo intelligent speaker. It can book a cab ride, play music by selecting the song from a playlist, store and remind the appointments from the calendar, read a book from kindle and audio bookstore on the device, and more. It is quicker to respond and understand commands with the Echo implemented with advanced, accurate voice recognition. Its functionality is not comparable with Siri, Google Now and Cortana. It is not good at responding complex queries

2.2 Digital Financial Trading and Investment Strategies

This section pertains to financial services including share trading, forex trading, investment management, etc. The digital world has made a significant impact on the traditional trading system with many new digital technologies. Algorithmic trading is one of them.

Algorithmic trading is also called algo-trading. It is the process of employing computer programs comprising instructions for issuing a trade to generate profit for the business with speed and frequency, which is impossible to do for any a human being. The predefined instructions are based on the timing of purchase or sell, price the count of stocks to trade or any mathematical model. The algo-trading provides an opportunity to increase the profit, and the trading becomes more systematic by ruling out the impact of human subjectivity emotion on trading activities [22].

Any rule-based trading strategy can be completely automated. The market data in the digital form is furnished to the rule-based trading models running inside an algorithmic trading system. Trading methodologies filter, process, analyze market data, and generate trading signals. Based on trading signals, actions are executed (e.g., submitting an order or terminating a position) and orders are directed to respective markets [23].

In digital financial domains, algorithmic trading means the utilization of algorithms or programs to automate one or more phases of the trading procedure like pretrade analysis of the data, trading signal generation, i.e., recommending for a buy or sell action, and finally trade consummation. Trade execution is additionally partitioned into agency/broker execution, i.e., when an algo-trading system optimizes issues signal for the trade on behalf of a client and principal/proprietary trading, i.e., when a business organization trades on its account. Each phase of this trading process can be carried out by human agents and trading-algorithms, or solely by trading algorithms.

Figure 6 depicts the major elements of an algorithmic trading system. It also depicts the stages at which they prevail in a trading process. The pretrade analysis encompasses three mathematical models:

- The alpha model, which predicts the upcoming performance of the financial instruments in the trade.
- The appraisal of the degree of exposure/risk correlated with the financial instrument is achieved by the risk model.
- The transaction cost model finds out the costs associated with the trading of the financial instruments.

Trading signal generation phase includes the portfolio construction model. This model accepts inputs the outputs of the previous phase viz., pretrade analysis encompassing alpha model, risk model, and transaction cost model. It carries out a decisive step with what portfolio of financial objects should be possessed while going forward and in what quantities. In the trade execution phase, the execution model carries out the trades, making several decisions with constraints on (actual) transaction costs and the duration for trade. The most common decision is the strategy building for trading followed by the venue and order type. The result of the execution model is analyzed in the real world context and the data again incorporated into the database for future analysis. In this way, the algorithmic trading system enhances its intelligence for future trading signal prediction [24].

There are several scientific articles involving algorithmic trading. Hu et al. reviewed 51 articles involving algorithmic trading utilizing evolutionary algorithms such as Genetic Algorithm (GA), Genetic Programming (GP), Learning Classifier System (LCS), and Particle Swarm Optimization (PSO) [25].

Seddon and Currie [26] developed a conceptual model for big data analytics to reap better benefits from high-frequency trading (HFT). The features of big data are categorized into big data, fast data, and big compute categories, and different priorities are accorded to the categories to produce the high performance of HFT. The work presents an understanding of algorithmic trading in global financial markets.

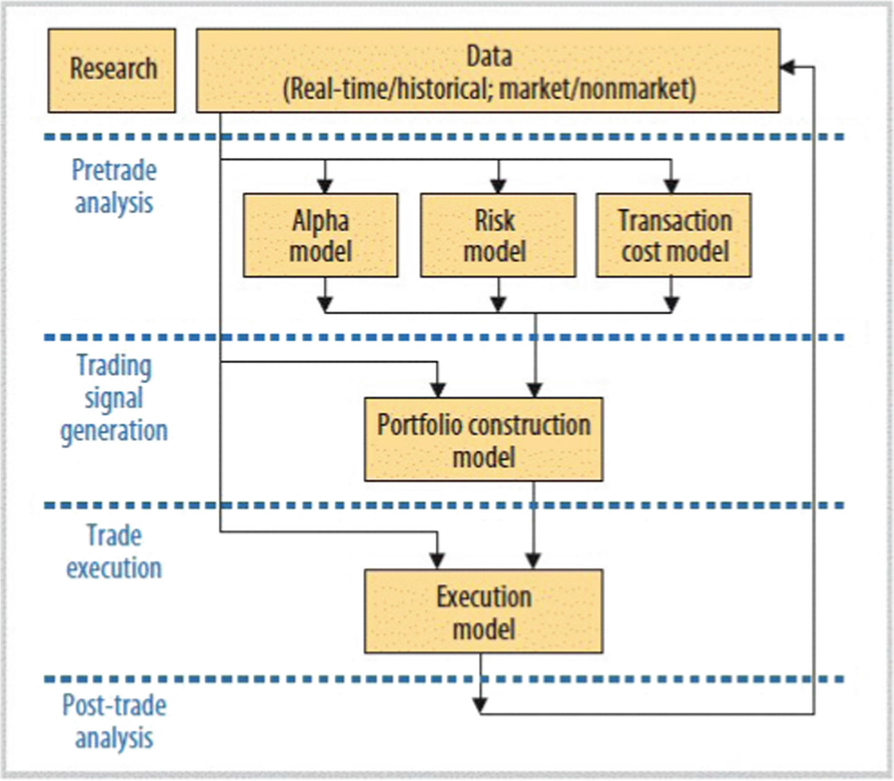


Fig. 6. Components of algorithmic trading system [24]

3 Futures of Digital Banking

The advanced, cutting-edge computer technologies, high-end computational power enabled digital devices with the state-of-the-art applications are proving their presence and influence in the modus-operandi of the banking industry and affecting the banking ecosystem. The most predominant leading-edge technologies include Big Data Analytics (BDA), Cloud Computing, Artificial Intelligence (AI) and Machine Learning (ML), Robotic Process Automation (RPA), blockchain and the Internet of Things (IoT).

The banking system or at a large the financial services industry is embracing new technologies and setting trends in that. The banks have used private clouds in the past years, but they will be moving towards the hybrid and public cloud implementation, which can lead to the development of agile applications. Financial institutions like, banks, payments, traders, credit card providers, insurance carriers have gathered a huge volume of historical data. These institutions have evolved themselves to handle real-time data. They can leverage on historical and real-time data by integrating operational and analytical system together to gain insights of the business value. One such example can be an algorithmic trading system, which uses both historical data of customer behavior and trading pattern as well as real-time data of trading. Also, the credit card transactions should include historical as well as real-time data to deal the fraud cases. The financial services are moving towards IoT and streaming for generating data from various devices embedded with sensors to generate real-time data and which is ingested to the system through streaming for real-time analytics [27].

3.1 Internet of Things (IoT) in Banking

In IoT, different types of sensors are embedded into the Internet-connected devices that gather data and share it over the internet with people, applications, and other devices. The ability to analyze the collected data to explore the business insights and apply the insights contextually has the potential to enhance the business of the banking industry.

The IOT devices can be any digital device, e.g., tablet, smartphone, wearable sensors or industrial sensors. It can send data to anyone over the network through any network path, e.g., wired or wireless. These IOT devices can be deployed anywhere for transmitting data related to many contexts employing the different services or applications related to different businesses.

The prominent players such as Apple, Google, Amazon, and Samsung have developed wearable gadgets and voice-first devices for personal assistance. A voice-first device is an always-on, intelligent piece of hardware where the primary interface is a voice for both input and output. This innovation has a continuous effort in the collection, analysis of the data for the application that can build consistent connections among solutions to augment the comfort level in a client's life. A brilliant example of this innovative technology can be the way Uber has amalgamated geospatial analytics, real-time pricing/demand analytics, and unified payment system to deliver a superior short-range transportation solution. The apex IoT platform providers for storage and analytics of the big data for insights are provided by Amazon Web Services (AWS), IBM's Watson, and Microsoft Azure.

Use Cases of IoT in Banking

The banking industry has started to reap the capabilities of IoT. A survey was conducted on a global scenario, and it was found that 64.5% of banking executives tracked their customers through the mobile apps used on different digital devices like smartphones, tablets and other digital devices where the apps can run. In addition to that, 31.6% of banking institutions utilized the IoT to observe retail locations, e.g., bank branches, 21.1% used sensors to collect product performance data and 15.8% executives employed IoT sensor embedded wearables to pursue customer product utilization [28].

Financial companies have started to give utmost priority to the customer, and product monitoring due to the increasing incidents of online and offline frauds. Identity verification has become a hectic task as identity theft also plays a major part in global fraud and the fright of computer system and network breach. Customers' financial transactions are tracked, and data are collected thereof by utilizing the IoT devices. Sensor data is also used for monitoring assets and evaluation of collateral while issuing loans.

The banking industry also uses IoT embedded in a wearable i.e., smartwatch or fitness band for basic banking.

Different research works have been carried out in BFSI sector employing IOT devices. Dineshreddy and Gangadharan [29] have proposed a framework for investment management using IOT.

3.2 Blockchain, AI and Its Role in Banking Industry

As we know, the Blockchain has come up as the next generation of payment channel without the involvement of any controlling third party. A Blockchain is a public ledger of all cryptocurrency (e.g., Bitcoin) transactions that have ever been executed. A Block is a unit of the Blockchain, which contains some or all of the recent transactions. It is the current part of a Blockchain, which once completed enters into the Blockchain as its permanent storage. The completed blocks are appended to the Blockchain in a linear, chronological order. The blocks are connected to each other by retaining the hash of the previous block. Each node in the Bitcoin network authenticates and broadcasts the transactions to the other nodes present in the network. The node receives a replica of the Blockchain, which is downloaded automatically upon connecting to the Bitcoin network. The Blockchain has comprehensive information such as the addresses and their balances starting with the inception block to the most recently appended block. Whenever, a block gets completed, a new block is generated [30].

The inclusion of ML to the Blockchain technology presents a new paradigm by providing Artificial Agents on the Blockchain technology rendering security and ensuring the immutability of all data.

It provides opportunities for financial firms to overhaul existing banking infrastructure and speed settlements. It immediately transfers funds securely, with no wait for confirmation. The Blockchain can handle electronic-Know Your Customer strategy, trade finance, cross-border payments, clearing and settling bond or equity trades. Thus, the combination of Blockchain technology with ML takes the banking technology to the next high level

The ICICI bank branch in Mumbai is the first Indian bank that made the cross-border payment through Blockchain payment to the largest bank of Dubai Emirates NBD. The banks have partnered with banking solution Infosys Finacle for this [31]. Reuters reports that several banks have announced plans to use Blockchain technology, with Microsoft teaming up with Bank of America [31].

3.3 Chatbots

A chatbot is a service or tool that you can communicate with via text messages. The chatbot understands what you are trying to say and replies with a coherent, relevant message or directly completes the desired task for you.

The services a Chabot can deliver are diverse. Important life-saving health messages, to check the weather forecast or to purchase a new pair of shoes, etc., are some example tasks they can perform.

The chatbot can talk to you through different channels; such as Facebook Messenger, Siri, WeChat, Telegram, SMS, Slack, Skype and many others.

The first chatbot ELIZA, created in 1966 by Joseph Weizenbaum [32] could recognize certain keywords and pattern and answer accordingly, mimicking conversation with a psychotherapist. The timeline of the development of chatbots is depicted in Fig. 7.

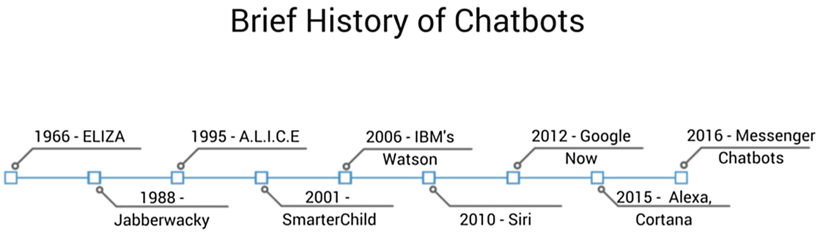


Fig. 7. Timeline of chatbot development [34]

The chatbots with AI derive their knowledge through machine learning and Deep learning running in the backend. The chatbots employ Natural Language Processing (NLP) in the frontend for interpreting the human language and then presenting it to the backend knowledge extraction process.

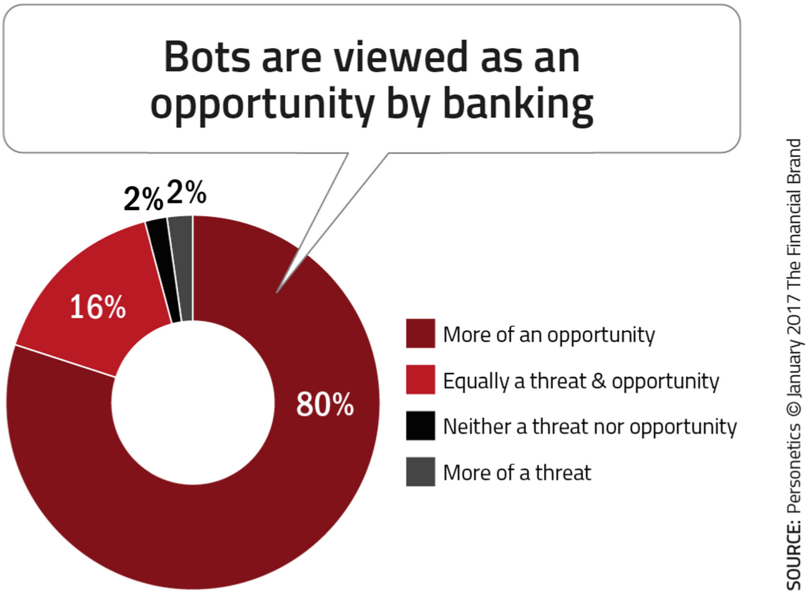
The chatbots can be employed in different fronts of business. They will come handy and reduce time, labor cost and increase efficiency leading to business value enhancement. In finance services industries we can employ a chatbot as an HR assistant, Market intelligence assistant, Workflow assistant, Social media channel assistant, Financial analyst assistant, Scheduling assistant and overall can be employed as the Brand ambassador for the business. These are the different dimensions of a business where usage of a chatbot can bring a paradigm shift.

According to a research study, 80% of global financial institutions regard chatbots as a golden opportunity to enhance business productivity. Only 16% view chatbots as both a threat and opportunity, whereas only 2% believe that chatbots could be a threat to the business being a loophole for breaching to vital business data (Fig. 8) [33].

Requirements of Intelligent Chatbots

The advanced intelligent based chatbots should fulfill the following eight requirements.

1. **Carry an intelligent conversation** – The chatbots should be equipped with good NLP components so that it can understand the context and any out of context statements like sarcasm and converse accordingly.
2. **Comprehend individual context** – An intelligent bot can comprehend each client's financial condition, along with current account holdings, the latest and expected financial behaviors, etc. to furnish financial advice and offers that are personalized in real time.
3. **Utilize real-time transactional data** – If the information presented by the bot is obsolete then the insights drawn from it will be inaccurate, clients will rapidly lose confidence in the information and stop further use of the bot. Therefore, chatbots should be designed such that they get access to the real-time transactional data in order to be relevant and current in replying.



SOURCE: Personetics ©January 2017 The Financial Brand

Fig. 8. Response to employment of chatbot in banking industry [33]

4. **Reuse existing content** – A chatbot should be able to draw the existing insights in real-time from different channels so that duplication of effort can be avoided.
5. **Be useful** – The chatbots should not be just efficient to tell the balance of an account. It should give some predictive and prescriptive insights to the queries of the customer from his contextual transactional data.
6. **Perform seamlessly across channels** – Clients anticipate a seamless experience across the entire digital channels viz. internet, mobile app, chatbots, virtual personal assistants. A conversation may start in Facebook Messenger, move to Amazon's Alexa, and continue in the bank's online or mobile app. Hence, the bot should be enabled with integration to Omni-channel strategy.
7. **Get smarter over time** – The bot must gather knowledge by continuous interaction with the client. The bot should be able to recollect and derive insight from the reactions made by the client when he/she was assisted with some information or advice and the feedback provided by the client. Thus, the bots will enhance their knowledge and deliver a personalized experience over time.
8. **Anticipate customer needs** – The usage data present that almost 50% of all bots are contacted only once and never again. This is certainly due to the shortcomings in the early implementations, discussed above. The bots should enthusiastically contact the right customers with the right information, at the right time with an individualized catering of the need based on predictive analytics [35].

The Future of Chatbots

The AI-enabled chatbots interact in real-time and assist with personalized and contextual answers to customer queries. It can inform the balance in the current account, transfer money to another account, pay a utility bill, and report recent spending activities.

It will be eventually equipped with the knowledge about the customer in fulfilling his/her requirement and also can look out for some better options and rewards for him/her based on the insights gathered from the deeper learning about him. Finally, the bot will influence with an Omni-channel presence by integrating with the virtual personal assistants like Facebook Messenger, Amazon's Echo, etc.

3.4 Robotics in Banking and Finance Industry

Robotics, empowered with AI (ML), is proving to be the most effective tool that can produce operational efficiencies to the entire financial services industry.

Robots come with unique advantages – they are time and cost efficient, improve productivity, deliver superior results, and can work without rest over repetitive tasks. When enabled with cognitive computing, AI (ML) capabilities, robots can be trained to operate autonomously. They can also learn how to improve performance and accuracy with little human input. Further, multi-lingual language processing and voice recognition capabilities allow robots to interact and conduct seemingly intelligent, coherent and meaningful conversations with customers.

For example, Bank of Tokyo- Mitsubishi UFJ (MUFG) introduced Nao, a 58-cm (1 ft 11)-tall, 5.4 kg robot developed by Aldebaran Robotics – a France-based subsidiary of Japanese telecom and internet giant SoftBank. It is equipped with a camera and microphone and has visual recognition and remote control capabilities. It can recognize 19 spoken languages, interact and communicate with customers in branches, and provide a response to queries [36].

The first banking robot of India is Lakshmi, launched by the Kumbakonam-based City Union Bank. It is the artificial intelligence powered robot which is the first on-site bank helper. Lakshmi can answer intelligently on more than 125 subjects [37].

All financial services companies operate in a highly regulated environment. They have to meet the demands of auditable, have security in a complete scenario, maintaining information-enriched data and operational resilience. Robotic Process Automation (RPA) allows modern banks to fulfill these requirements and achieve high operational efficiency.

Besides the cost savings, efficiency improvements through higher productivity, ability to work 24 × 7, and greater accuracy by reducing human errors can be achieved. Their ability to collect and mine vast data and complete audit ability are especially useful in areas like compliance and regulatory reporting. Furthermore, these robots can be deployed and scaled with ease and agility. Utilizing the RPA the financial service institutions can achieve the major activities those were otherwise human error prone with the human interface. The robotic skills those earlier relied upon human skills, and manual effort only are as follows.

- Collecting, collating and validating the data from the customers.
- Synthesizing, and analyzing the structured and unstructured data.
- Calculations and based on the result decision-making capability.
- Communications with NLP and assisting clients and customers.

- Orchestrate and manage both robotic and people-based activities.
- Monitor, detect and report the operational activities.
- Learning, anticipating, and forecasting behaviors and results.

4 Insurance and Analytics

Insurance, another sub-domain of the financial services, generates a huge amount of data, which can be analyzed by employing descriptive, predictive and prescriptive analytics to derive business insights for enhancing business while rendering better service to their clients.

4.1 Descriptive Analytics and Insurance

Descriptive analytics provides the results that further help the organization to enhance their benefit. This type of analytics uses the historical data present with the organization such as weblog, interactions on social media, interaction with chatbots, online feedbacks, and sensor data and so on.

For instance, a sensor device is attached to the car, which will monitor different activities such as total amount of distance traveled, a sudden change in speed, duration of high-risk driving, and number of the time hard brakes are applied. These events will be treated as predictor variables while pricing the insurance premiums.

4.2 Predictive Analytics and Insurance

Using predictive analytics, insurance companies can predict trends in various activities. For instance, predicting possible fraudulent claims, various health care frauds, insurance needs based on customer data etc. Thus, predictive analytics brings in revenue for the company.

4.3 Prescriptive Analytics and Insurance

Prescriptive analysis suggests some optimized solutions to a problem based on other analytical results. Prescriptive analytics is always evolving and of crucial importance to the business values.

Example of prescriptive analytics are as follows: (i) Casualty insurance providers can use historical climatic data for catastrophe modeling and prediction thereby recommending the right price for the insurance. (ii) It can present policy conditions and optimize portfolios to keep a check on the rise of risk. (iii) Recommending optimal price plans, establish policy conditions, and optimize portfolios to keep the accumulation of risk in check. Prescriptive analytics empowered by data can shape the future of the insurance industry [38].

A lot of research has been carried out for boosting the insurance industry. The insurance customer profitability is the benefit that the insurance company is going to receive based on the premium profit and claim risk for the customer. Fang et al. [39] implemented random forest for forecasting insurance customer profitability using big data analytics which turned out to be superior to traditional forecasting methods, such as linear regression, decision tree, and SVM.

Big data analytics plays a major role in the insurance industry in its different aspects. The insurer performs risk analytics for issuing any insurance. Predictive analytics can be used to fix the premium value of vehicle insurance by analyzing the historical data of the driver, i.e., predicting the probability of the driver to meet an accident. Similarly, in case of health insurance, the wearable IoT gadgets involving health-monitoring sensors, gathers historical health data of the customer, which can predict the customer profitability to the insurer. For claim fraud detection, the claims are matched against the profiles and with the past claim patterns which were fraudulent or those that are deviating from the pattern of genuine claims. This may involve the behavior of the claimant, the people connected with him through his social network and the agencies involved in the claim. Customer insights can be gained from interactions through multiple channels. This can help provide relevant product and identify the right segment for up-selling/cross-selling [40].

A brand community is a group formed based on attachment to a product or brand. This also represents highly valuable marketing, innovation management, and CRM tools. The brand communities bring the brand and community together to a single platform. The social interaction among the members of the community members influences the customer relationship and perspective towards the product. Hence, the social network analytics will result in enhancement of the business value for the financial service industries. Zaglia [41] presented how brand communities are embedded in social networks and how to leverage it.

5 Discussion

The big data analytics is changing the face of financial services industry. In the banking industry, huge amount of system-generated and customer-generated data propels of growth of the business. The following are the different ways that big data analytics can bring forth a new face to the banking industry. Various forms of analytics discussed in the paper indeed can benefit immensely from the big data available in the respective business/operational problems including the new technologies IoT, Blockchain, Robotics, Chatbots and data lake.

- The customer analytics provide a 360° view of a customer to make a proper decision for personalized marketing.
- The risk analytics helps to determine credit score of a customer to take decisions on granting a loan.
- The social analytics provides the insights for cross-selling also it can help in preventing frauds.
- The analysis of customer interaction over multiple channels can help the bank to present some personalized offers.
- The sensor data from IoT devices can be analyzed for a better understanding of the customer behavior pattern.

- The online footprints on the bank's website and spend pattern analysis will provide the insight for cross-selling.
- The banks can also analyze the offline interaction of the customer with the bank through the ATM data, credit or debit card transaction data.
- The data from online and offline interaction with the bank can be analyzed for churn prediction, market basket analysis, increasing the customer life period.
- The security analytics helps to provide a fraud-free environment, which helps in building the brand community.

Despite the quintessential benefits of data analytics/big data analytics in the financial services industry, a lot of challenges still remain, which impede its full-scale implementation: Lack of

- good data quality is the biggest hindrance in implementing BDA.
- analytical-savviness in an organization is an equally important roadblock.
- qualified and well-trained manpower is another stumbling block.
- participation of the business or user departments stymies the implementation of BDA.
- executive support in the makes the BDA implementation a non-starter.
- phased approach in implementation
- change management planning in terms of mindset change across organization

These can be considered critical success factor for the BDA implementation.

6 Conclusions

The paper presented the current digital trends in financial services industry with a particular emphasis on the banking industry, where several business problems solved by big data analytics are highlighted. It also covered some state-of-the-art technologies, which together with big data analytics brought a substantial change in the quality and productivity of the banking industry. The stock market and insurance disciplines were also explored, where predictive and prescriptive analytics play a paramount role. This paper concludes with some challenges, which impede full-scale implementation of big data analytics in financial services industry to enhance the business value.

References

1. Why only one of the 5 Vs of big data really matters. IBM Big Data & Analytics Hub. http://www.ibmbigdatahub.com/blog/why-only-one-5-vs-big-data-really-matters
2. Apache Hadoop. https://en.wikipedia.org/wiki/Apache_Hadoop
3. Apache Spark Introduction. https://www.tutorialspoint.com/apache_spark/apache_spark_introduction.htm
4. Krishna, G.J., Ravi, V.: Evolutionary computing applied to customer relationship management: a survey. Eng. Appl. Artif. Intell. **56**, 30–59 (2016)
5. Erevelles, S., Fukawa, N., Swayne, L.: Big data consumer analytics and the transformation of marketing. J. Bus. Res. **69**, 897–904 (2016)

6. Kshetri, N.: Big data's role in expanding access to financial services in China. Int. J. Inf. Manage. **36**, 297–308 (2016)
7. Sun, E.W., Chen, Y.-T., Yu, M.-T.: Generalized optimal wavelet decomposing algorithm for big financial data. Int. J. Prod. Econ. **165**, 194–214 (2015)
8. Express Computer, vol. 28, No. 8. By Indian Express – Issuu, August 2017. https://issuu.com/indianexpressgroup/docs/ec-201708pages
9. Akter, S., Wamba, S.F., Gunasekaran, A., Dubey, R., Childe, S.J.: How to improve firm performance using big data analytics capability and business strategy alignment? Int. J. Prod. Econ. **182**, 113–131 (2016)
10. Social network analysis for fraud detection in payments - Banking.com. Banking.com. http://banking.com/analysis/social-network-analysis-for-fraud-detection-in-payments/
11. Kirchner, C., Gade, J.: Implementing social network analysis for fraud prevention. CGI Gr. Ind. (2011)
12. Mitigating and detecting financial fraud with social network analysis and graph database. https://resources.zaloni.com/blog/mitigating-detecting-financial-fraud-with-social-network-analysis-and-graph-database
13. Chau, D.H., Faloutsos, C.: Fraud detection using social network analysis, a case study. In: Alhajj, R., Rokne, J. (eds.) Encyclopedia of Social Network Analysis and Mining, pp. 547–552. Springer, New York (2014). https://doi.org/10.1007/978-1-4614-7163-9_284-1
14. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Understanding belief propagation and its generalizations. Explor. Artif. Intell. New Millenn. **8**, 236–239 (2003)
15. Saidi, F., Trabelsi, Z., Salah, K., Ghezala, H.: Ben: approaches to analyze cyber terrorist communities: survey and challenges. Comput. Secur. **66**, 66–80 (2017)
16. Kumar, B.S., Ravi, V.: A survey of the applications of text mining in financial domain. Knowl. Based Syst. **114**, 128–147 (2016)
17. West, J., Bhattacharya, M.: Intelligent financial fraud detection: a comprehensive review. Comput. Secur. **57**, 47–66 (2016)
18. Digital Forensics. https://en.wikipedia.org/wiki/Digital_forensics
19. What is Digital Forensics? InterWorks, Inc. https://www.interworks.com/blog/bstephens/2016/02/05/what-digital-forensics
20. Xu, S.: Cybersecurity dynamics: a foundation for the science of cyber security. http://www.cs.utsa.edu/~shxu/socs/
21. Digital banking framework (2016). http://www.idrbt.ac.in/assets/publications/BestPractices/DigitalBankingFramework_Nov2016.pdf
22. Basics of algorithmic trading: concepts and examples. Investopedia. http://www.investopedia.com/articles/active-trading/101014/basics-algorithmic-trading-concepts-and-examples.asp
23. Algorithmic trading software – AlgoTrader. http://www.algotrader.com/
24. Nuti, G., Mirghaemi, M., Treleaven, P., Yingsaeree, C.: Algorithmic trading. Computer (Long. Beach. Calif) **44**, 61–69 (2011)
25. Hu, Y., Liu, K., Zhang, X., Su, L., Ngai, E.W.T., Liu, M.: Application of evolutionary computation for rule discovery in stock algorithmic trading: a literature review. Appl. Soft Comput. **36**, 534–551 (2015)
26. Seddon, J.J.J.M., Currie, W.L.: A model for unpacking big data analytics in high-frequency trading. J. Bus. Res. **70**, 300–307 (2017)
27. Top 10 big data trends in 2017 for financial services. MapR. https://mapr.com/blog/top-10-big-data-trends-2017-financial-services/
28. Should banking build an Internet of Things (IoT) strategy? https://thefinancialbrand.com/63285/banking-internet-of-things-iot-data-analytics-payments/

29. Dineshreddy, V., Gangadharan, G.R.: Towards an “Internet of Things” framework for financial services sector. In: 2016 3rd International Conference on Recent Advances in Information Technology (RAIT), pp. 177–181. IEEE, Dhanbad (2016)
30. Blockchain. http://www.investopedia.com/terms/b/blockchain.asp
31. What is “blockchain” and why ICICI Bank’s use of it is a big deal? VCCircle. https://www.vccircle.com/what-blockchain-and-why-icici-bank-s-use-it-big-deal/
32. Story of ELIZA, the first chatbot developed in 1966. http://analyticsindiamag.com/story-eliza-first-chatbot-developed-1966/
33. Banks and Credit Unions bullish on chatbots for customer service. https://thefinancialbrand.com/63596/financial-banking-bots-chatbot-voice-ai/
34. Best practices for building chatbots and conversational interfaces. https://www.altexsoft.com/blog/business/a-comprehensive-guide-to-chatbots-best-practices-for-building-conversational-interfaces/
35. 8 things your bot should do to make customers smile – Personetics. Personetics. https://personetics.com/resource_center/8-things-bot-make-customers-smile/
36. How robots are changing the face of banking. The Asian Banker. http://www.theasianbanker.com/updates-and-articles/how-robots-are-changing-the-face-of-banking
37. Lakshmi, country’s first banking robot, makes debut in Chennai - Times of India. https://timesofindia.indiatimes.com/city/chennai/Lakshmi-countrys-first-banking-robot-makes-debut-in-city/articleshow/55361225.cms
38. 4 types of analytics defining the future of the insurance industry. Vertafore. http://www.vertafore.com/Resources/Blog/4-Types-of-Analytics-Defining-the-Future-of-the-Insurance-Industry
39. Fang, K., Jiang, Y., Song, M.: Customer profitability forecasting using Big Data analytics: a case study of the insurance industry. Comput. Ind. Eng. **101**, 554–564 (2016)
40. Top 7 big data use cases in insurance industry — Exastax. https://www.exastax.com/big-data/top-7-big-data-use-cases-in-insurance-industry/
41. Zaglia, M.E.: Brand communities embedded in social networks. J. Bus. Res. **66**, 216–223 (2013)

A Framework to Improve Reuse in Weather-Based DSS Based on Coupling Weather Conditions

A. Mamatha[1](✉), P. Krishna Reddy[1], Anirban Mondal[2], Seishi Ninomiya[3], and G. Sreenivas[4]

[1] Kohli Center on Intelligent Systems (KCIS), IIIT Hyderabad,
Hyderabad, Telangana, India
mamatha.a@research.iiit.ac.in, pkreddy@iiit.ac.in

[2] Shiv Nadar University, Greater Noida, Uttar Pradesh, India
anirban.mondal@snu.edu.in

[3] University of Tokyo, Tokyo, Japan
snino@isas.a.u-tokyo.ac.jp

[4] Professor Jayashankar Telangana State Agricultural University, Hyderabad, India
gsreenivas2002@gmail.com

Abstract. In weather-based decision support system (DSS), the domain experts provide suggestions to carry out appropriate measures to improve the efficiency of the respective domain by analyzing both the forecasted and observed weather values. In this paper, to provide suggestions for a given combination of forecasted and observed values, we have proposed a framework to exploit *reuse* of the suggestions which have been prepared for the past combinations of observed and forecasted values over the years. We define the notion of coupled weather condition (CWC) which represents the weather conditions of two consecutive durations for a given combination of weather variables. By employing the domain-specific categories, the proposed framework exploits the reuse of CWCs for the given domain. We have applied the proposed framework by considering the case study of agromet advisory service of India Meteorological Department (IMD). The extent of reuse has been computed by considering 30 years of weather data from Rajendranagar, Hyderabad, Telangana State, based on the weather categories data provided by IMD. The reuse over 30 years is computed by considering the period of year and crop seasons of a year. Period is defined as portion of time of the year(s) that is considered to analyze the similarity. The results are very positive. The results show that the percentage of reuse of CWCs with three weather variables for the period of year is about 77% after five years. The results provide the scope to develop automatic weather-based DSS in various domains with minimal human intervention and improve the utilization of the generated content.

Keywords: Reuse · Decision support systems · Weather patterns
Agromet service · Agro-informatics · Data mining

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 40–58, 2017.
https://doi.org/10.1007/978-3-319-72413-3_3

1 Introduction

Weather is the state of the atmosphere as measured on a scale of hot or cold, wet or dry, calm or storm, and clear or cloudy [9,17]. Weather is typically represented by means of the values of weather variables. As weather has a significant impact on several important and diverse domains such as agriculture, health and transportation, governments are nowadays investing huge amounts of budget for operating weather forecasting systems.

A decision support system (DSS) is an interactive computer-based information system that supports decision-making in various domains. Over the past few decades, dedicated efforts have been made to build efficient computer-based automatic DSSs to improve the efficiency of various domains. The notion of *reuse* has been widely exploited to build efficient DSS in medical and software engineering domains [15,16]. In a similar vein, weather-based DSSs are also being built to improve the efficiency of the production systems in domains such as healthcare, agriculture and live-stock, transportation, business and financial planning, governance, etc.

Systems that operate on weather consider both past weather information (observed) and future weather prediction (forecasted) information to analyze the impact of weather on the given domain. Based on the values of observed and forecasted weather values, the weather-based DSS provides appropriate suggestions and recommendations to the relevant stakeholders in the given domain. Exploiting the notion of *reuse* in weather-based DSSs could facilitate in improving both its efficiency as well as its scalability.

Incidentally, the concept of *duration* plays an important role in weather-based DSSs. Duration is defined as the time span during which the values of weather variables are recorded. For example, weather data are available for sub-hourly, hourly, daily, monthly and annual durations. The daily values are computed from sub-hourly and hourly recorded data. Similarly, weekly, monthly and annual values are derived from the daily data. For the given duration, a combination of a given set of the values of the weather variables is referred to as a *weather condition (WC).* To analyze the reuse of observed and forecasted values, we introduce the notion of *coupled weather condition (CWC)* which represents the weather conditions of two consecutive durations over the year. The labels assigned for the range of each weather variable are referred as *domain-specific categories.* These categories indicate how the values of the weather variables impact a given domain.

By employing the domain-specific categories, the proposed approach would analyze the extent of *reuse* of CWC in any given domain. We have conducted the experiments on 30 years of weather data from Rajendranagar, Hyderabad, Telangana State, based on the weather categories data provided by India Meteorological Department (IMD). By varying the number of weather variables in CWC from one to five with durations equal to 1 day and 5 days respectively, we have computed the extent of similarity between the periods of the year and season-wise CWCs of rice crop i.e., summer, kharif and rabi. The experiment results indicate that there is a significant degree of similarity among CWCs over the years. In particular, for CWCs with three variables about 73% of *reuse* can be observed within two to three years. Thus, the results of our experiments demonstrate that there is a scope to develop automatic weather-based DSSs with minimal human intervention and also there are opportunities for improving the utilization of the developed content such as standardized suggestions, recommendations and advisories.

The major contributions of the paper are as follows.

(i) A knowledge pattern called coupled weather condition (CWC) is proposed to capture a weather phenomena which is a combination of observed and forecasted weather values of set of weather variables for a given duration.
(ii) A generalized framework is proposed to exploit reuse in weather-based DSS for any domain by exploiting domain-specific categories.
(iii) Applied the proposed framework on 30 years of weather data for the periods of year and seasons (summer, kharif and rabi) and analysed the percentage of reuse by considering the case study of agromet advisory service of India Meteorological Department.

The remainder of the paper is organized as follows. Section 2 presents the Related work, while Sect. 3 describes the proposed framework. Section 4 presents the case study as well as the experimental results. Finally, we conclude in Sect. 5 with directions for future work.

2 Related Work

One of the key ways to improve the efficiency of software development is to improve the “reuse”. In this regard, the work in [15] proposed a framework to prove that reuse indeed improves the performance of software project management. Moreover, the work in [14] showed that reuse also improves the efficiency of large-scale systems.

The DSSs solve many semi-structured and unstructured problems and facilitate towards making decisions in environments, where the underlying circumstances may change rapidly, thereby necessitating dynamic decision-making capabilities. The work in [4] discussed the utilization and technology issues associated with DSSs in the context of large-scale complex systems. In particular, it indicated that DSSs are able to address the complexity of real-world decision-making problems by increasing the efficiency in the production environment. Notably, DSSs play a significant role in the healthcare domain. The capture and reuse of ICT-based knowledge in clinical decision support system (CDSS), which add value to medical diagnosis, have been discussed in [16]. Furthermore, clinical reminder systems have been becoming increasingly popular. Clinical reminder systems manage the workflow of the health care domain and serve as a decision support system to generate timely reminders pertaining to patient care [5].

Many decisions in our day-to-day lives w.r.t. various domains (ranging from transportation to agriculture) are typically influenced by weather. As an example, in the transportation domain, the work in [2] proposed an approach to investigate the impact of weather on travel time prediction. In a similar vein, for the agricultural domain, the work in [7] presented a study on the economic impact of Agro-Advisory System (AAS) Versus non-AAS in cultivation and production. In particular, the aim of the study was to analyze the statistics of the percentage of increase in yield using AAS. Furthermore, a weather-based DSS, designated as PROPLANT [13], has been built for controlling fungal diseases in winter wheat crops. The impact of weather on sustainable agriculture has been discussed in [10].

In agriculture-based DSSs, weather-based advisory decisions can be improved by means of accurate weather forecasts. The work in [11] proposed a framework for advice preparation and dissemination using IT-based agro-meteorological system (eAgromet). In a similar vein, the proposal in [6] discussed a system for efficient crop management based on improved weather forecasts. Moreover, the proposal in [3] discussed reuse in agro-advisories by using the notion of weather window for dominant crops. Furthermore, the work in [12] proposed a framework for improving the practical agricultural education using the notion of virtual crop labs.

By considering the weather categories provided by India Meteorological Department [1], an effort [8] has been made to propose a framework for computing the extent of similarity among the weather conditions of a given year and its preceding years. In this paper, we have defined the notion of *coupled weather conditions (CWCs)* to improve the performance of DSSs based on weather forecast. Additionally, we have proposed a generalized approach and demonstrated through experimental results that there is a scope to exploit reuse among CWCs in weather-based DSSs of any domain.

3 Proposed Framework

The weather is represented by the values of weather variables. The examples of weather variables are rain fall (RF), maximum temperature (Tmax), minimum temperature (Tmin), maximum relative humidity (RHmax) and minimum relative humidity (RHmin). The sample weather values for 7 days of Rajendranagar weather station for the year 2015 are given in Table 1.

Note that there are several other weather variables like wind speed, wind direction, cloud cover, etc. In this paper, we explain the proposed framework by considering preceding five weather variables only. However, the proposed framework is a generic approach and can be applied for other weather variables too.

3.1 Basic Idea

The weather situation of given duration plays a major role in planning many decisions related to various domains. Normally, domain expert analyses the weather values of observed and forecasted weather data, and then provides appropriate suggestions. For a given domain, if we develop content which contains domain-specific decisions for different weather situations, there is a possibility to *reuse*

Table 1. Sample daily weather data from 1^{st} March 2015 to 14^{th} March 2015, collected at Rajendranagar weather station, Hyderabad, Telangana State, India. The units of rain fall, temperature, humidity are millimeter (mm), degree centigrade (°C), percent (%) respectively.

Date	RF	Tmax	Tmin	RHmax	RHmin
1-Mar-2015	0.0	33.0	17.0	85.0	58.0
2-Mar-2015	8.0	28.0	20.4	83.0	43.0
3-Mar-2015	0.0	31.0	19.5	75.0	34.0
4-Mar-2015	0.0	32.0	16.0	65.0	33.0
5-Mar-2015	0.0	32.0	18.2	61.0	44.0
6-Mar-2015	0.0	33.0	21.9	82.0	62.0
7-Mar-2015	13.0	32.0	19.5	91.0	44.0
8-Mar-2015	8.6	32.0	19.5	84.0	40.0
9-Mar-2015	0.0	32.0	19.0	75.0	63.0
10-Mar-2015	0.0	28.5	20.5	69.0	58.0
11-Mar-2015	0.0	28.5	16.0	52.0	26.0
12-Mar-2015	0.0	32.5	15.0	56.0	31.0
13-Mar-2015	0.0	32.5	16.0	82.0	21.0
14-Mar-2015	0.0	34.0	18.5	38.0	13.0

the suggestions based on the occurrence of similar weather situations, thereby providing an opportunity to improve the performance of DSSs. The following concepts are being proposed to exploit the reuse in weather-based DSS: weather condition (WC), coupled weather condition (CWC), category-based coupled weather condition (CCWC). We also present the methods to compute the similarity among two WCs, CWCs, and CCWCs. Here, WC is the summary of the statistics of weather variables and CWC is the combination of two WCs. As CWC contains numeric values as weather statistics, it is difficult to find similar CWCs for a given CWC. So, we have employed the domain specific categories and defined the notion of CCWC. In CCWC, the numeric values are replaced with the corresponding category. By using the notion of CCWC, there is a scope to find similar CCWCs for a given CCWC from a given set of CCWCs over the years.

Now we explain the approach by explaining concepts of WC, CWC, and CCWC with motivation.

The notion of weather condition for a given duration d which is as follows.

Definition 1. Weather Condition (WC): The weather condition triple $\langle s_i, d, V\rangle$, where s_i indicates the start date, d indicates the duration equal to the number of subsequent days including s_i and V is the set of weather variables and the value of each weather variable equals to the summary statistics for d.

If ($d > 1$), the statistics for each weather variables in $\langle RF$, $Tmax$, $Tmin$, $RHmax$, $RHmin\rangle$ are to be calculated. Note that appropriate function should be employed to compute the statistics for $Tmax$, $Tmin$, $RHmax$ and $RHmin$ the value is equal to the mean value over d days whereas for RF, the value represents the cumulative value over the d.

Example: Weather condition with $d = 1$ and the start date "1 March 2015" and $V = \{RF, Tmax, Tmin, RHmax, RHmin\}$ is $\langle$1 March 2015, 1, $\{$0.0, 33.0, 17.0, 85.0, 58.0$\}\rangle$. Also, the weather condition with $d = 5$ and the start date "1 March 2015" is $\langle$1 March 2015, 5, $\{$8, 31.2, 18.22, 73.8, 42.4$\}\rangle$.

Based on the combination of observed and forecasted values for the given (current) date, appropriate suggestion is given in weather-based DSS. Here, we define the notion of coupled weather condition (CWC) to represent the combination of observed and forecasted weather values by combining two weather conditions.

Definition 2. Coupled Weather Condition (CWC): Consider two weather conditions $p = WC(s_i,\, d,\, V)$ and $q = WC(s_j,\, d,\, V)$. Also, $s_j = s_i + 1$. Then, the pair $\langle p, q\rangle$ is called coupled weather condition.

From Table 1, for given $d = 1$ and V $= \{RF, Tmax, Tmin, RHmax, RHmin\}$, the CWC(1 Mar 2015, 2 Mar 2015, 1, V) is given as $\langle$(((0.0, 8.0),(33.0, 28.0), (17.0, 20.4), (85.0, 83.0), (58.0, 43.0)$\rangle$ which represents two consecutive durations (s_i, s_j) i.e., (1 Mar 2015, 2 Mar 2015) weather values of each weather variables in V.

Now, we define the notion of period which is used to analyze the similarity of two CWCs.

Definition 3. Period (p): The notion of *period* is similar to duration and indicated by number of days. Normally, the number of days in the *period* is larger than a duration. For example, *period* can be year or season.

We compute the similarity between the CWCs of same period or different periods. The criteria to compare two CWCs are defined as follows.

Definition 4. Similar CWCs: Let x and y be the identifiers of two CWCs of different periods or the same period with V weather variables and having d duration for each CWC of x and y, i.e., $x = CWC(p, q)$ and $y = CWC(a, b)$. Let $sim(x, y)$ indicate the similarity of CWC. We say $sim(x, y)$ is equal to 1 if the values of corresponding weather variables are equal. Otherwise, $sim(x, y)$ is equal to 0.

The CWC represents the numerical values of the weather variables. But it is not possible to achieve high similarity among these CWCs as the similarity value comes to zero even with a small difference in any one variable value among the two CWCs. However, it can be observed that in several domains like medical domain or agro-meteorology domain, a different suggestion or advice is not recommended for a small change, like 0.2°C in temperature value or small change like 2% in humidity value.

Based on the preceding observation, we employ the category and define the similar CWCs. Note that it is assumed that for any domain appropriate categories are defined for each weather variable based on the influence of weather variable on that domain. For example, in the domain of agriculture, the variable rainfall is divided into several categories like light rain, moderate rain, heavy rain and so on.

Given the categories of each weather variable, we now define the term *category-based coupled weather condition.*

Definition 5. Category-Based Coupled Weather Condition ($CCWC$): Consider that the domain of each variable is divided into a set of categories such that each value of the weather variable is mapped to an appropriate category. The CCWC is a CWC in which the value of each weather variable is replaced with the corresponding category.

We now define similar CCWCs.

Definition 6. Similar CCWCs: Let x and y be the identifiers of two CCWCs of different periods or the same period with V weather variables and having d duration for each CCWC of x and y, i.e., $x = CCWC(p, q)$ and $y = CCWC(a, b)$. Let $sim(x, y)$ indicate the similarity of CCWC. We say $sim(x, y)$ is equal to 1 if the category values of corresponding weather variables are equal. Otherwise, $sim(x, y)$ is equal to 0.

Example: The CCWCs for a given five weather variable can be represented as the CCWCs for 1 variable, 2 variables, 3 variables, 4 and 5 variables of given weather condition. For given $d = 1$, CCWCs of $RHmax$ for two consecutive days i, j represented as $CCWC_1(i, j, 1, \langle RHmax \rangle)$, $CCWC_1$ represents 1 variable coupled weather condition represented as $\langle$(Low, High)$\rangle$ or $\langle$(High, Very High)$\rangle$. Here there is a change in the weather category of given CCWC.

3.2 Framework

Based on the concepts presented in the preceding section, we propose the framework to compute the similar weather conditions for a given weather condition over a given period. Let n be the current year and p be the given period, We assume that given the value of $(n > 1)$, daily weather data for V weather variables is available for $(n-1)$ years. Based on the domain requirements, the value of period p and duration d (in number of days) is determined. It is also assumed that the categories for the given weather variables for the given domain are given.

The proposed framework is divided into two parts: Computing CCWCs of the given period for $(n-1)$ years and computing similar $CCWCs$ for n^{th} year $CCWC_i$, where $CCWC_i \in$ CCWCs of given period for n^{th} year. In the first part we compute CCWCs of the given period for $(n-1)$ years. In the second part, the system receives $CCWC_i$ $(i > 0)$ for the current year (n^{th} year). Then, the system computes CCWCs of given period from $(n-1)$ years that are similar to each of the given $CCWC_i$, where $CCWC_i \in$ CCWCs of given period for nth year. The framework is depicted in Fig. 1.

I. Compute CCWCs for (n−1) Years: This step consists of three sub-steps.

(i) Computation of WCs: The input to this step is daily weather data of the given period for past $(n-1)$ years, where $n \geq 2$ with V weather variables and duration d i.e., required duration for the application domain. By considering starting date of each year as s_i, the WCs triples $\langle s_i, d, V \rangle$ are calculated for $(n-1)$ years.

(ii) Computation of CWCs: The input to this step is WCs of the given period for $(n-1)$ years. The corresponding CWCs are calculated for each year.

(iii) Computation of CCWCs: The input to this step is CWCs and domain specific categories for given weather variables. The corresponding CCWCs are calculated for each year.

II. Compute Similar CCWCs for Given CCWC$_i$: This step consists of two sub-steps.

(iv) Computation of CCWC for n^{th} ***year:*** The input to this step is CWC_i of given period for n^{th} year, duration d and domain specific categories. The domain expert give weather data of the given period for n^{th} year. In this algorithm, the value of period is considered as less than a year. By using the domain specific categories, the corresponding $CCWC_i$ is computed.

(v) Computation of similar CCWCs for $CCWC_i$**:** The input to this step is CCWC$_i$ and CCWCs of the given period for $(n-1)$ years. The system extracts CCWCs similar to CCWC$_i$ using the similarty criteria proposed in Definition 6. The pseudo code to extract the similar CCWCs of each year is presented in Algorithm 1.

Algorithm 1. Compute Similar CCWC

```
INPUT $CCWC_i$ of given period (p) for $n^{th}$ year, $CCWCs$ of given period (p) for $(n-1)$ years
OUTPUT Similar CCWCs for $CCWC_i$
initialize $S_1 = CCWC_1$;                ▷ CCWCs for $1^{st}$ year
initialize $SimCCWC = [\ ]$               ▷ List to store similar CCWCs for given $CCWC_i$
for $j \in \{2, ..., n\}$ do              ▷ $n$ represents number of years
    $S_j = S_{(j-1)} \cup CCWC_j$;
end for
$SimCCWC = \mid CCWC_i \cap S \mid$;
return $SimCCWC$;
end
```

In the remaining part of the paper, we use the notation "CWC" instead of "CCWC", as we perform all our experiments on categorical CWCs. We evaluate the performance of the proposed approach by considering a case study of agromet advisory service in India.

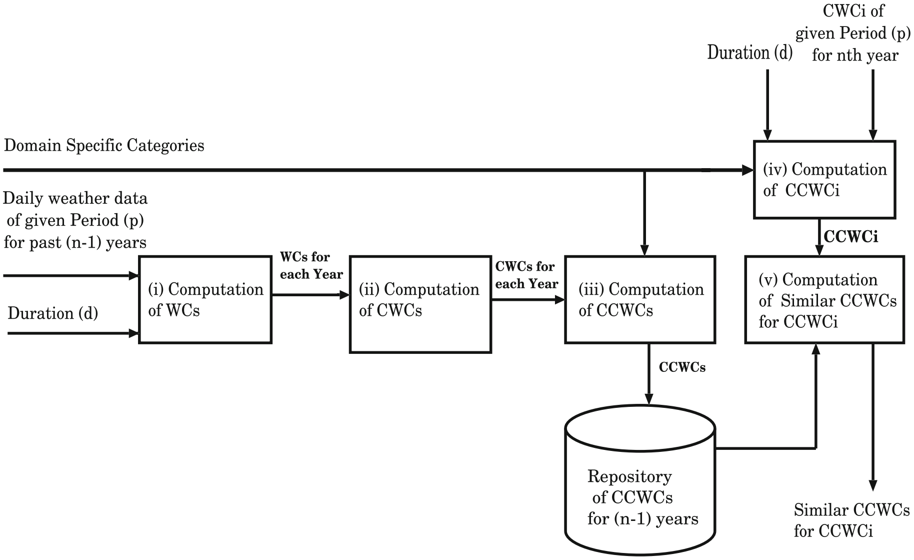

CWCi of
given Period (p)
for nth year
Duration (d)
Domain Specific Categories
(iv) Computation of CCWCi
Daily weather data of given Period (p) for past (n-1) years
CCWCi
WCs for each Year
CWCs for each Year
(i) Computation of WCs
(ii) Computation of CWCs
(iii) Computation of CCWCs
(v) Computation of Similar CCWCs for CCWCi
Duration (d)
CCWCs
Repository of CCWCs for (n-1) years
Similar CCWCs for CCWCi

Fig. 1. Architecture of proposed framework

4 Case Study: Agromet Advisory Service of India Meteorological Department

Weather and climate information play a major role before and during the cropping season and if provided in advance can help farmers apply resources in order to take advantage of favorable conditions and mitigate potential losses in unfavorable ones. We first explain about the Integrated Agro-Meteorological Advisory Service (IAAS) of India Meteorological Department (IMD) [1]. Next, by considering IAAS service framework, we will explain the experimental setup.

4.1 Integrated Agromet Advisory Service by IMD

In an environment of increasing weather and climate variability under climate change, farmers are in greater need of agro-meteorological information blended with weather sensitive management advisories before the start of cropping season to support the adaptation of agricultural practices.

In India, IMD issues various range of weather forecast that helps in planning measures for crop protection and management using the agro advisory system. IMD is currently providing forecasting services to farmers, fishermen, shipping, air navigation etc. IMD has started weather services for farmers from the year 1945. From the year 2008, IMD also started IAAS in collaboration with different organizations, for the duration of 5 days weather forecast.

In IAAS, after receiving weather forecast for a given region, the agromet experts prepare the agromet bulletin in local language based on the weather forecast, observed weather values of that region, crop stage and crop status. The agromet advisory bulletin contains the information on weather and weather-based advisories for crops and livestock which will be disseminated to farmers and related stakeholders. In this connection an effort has been made to help agromet scientists in the preparation of agromet advisory using eAgromet [11]. eAgromet is an IT-based agro advisory system, it is a web-based application that helps agromet scientists in the preparation and dissemination of agromet advisory.

4.2 Experimental Setup

In this section, by applying the proposed framework on the weather data set of 30 years, we analyze the extent of similar weather conditions by considering the case study of the agricultural domain. We analyze the similar CWCs with $d = 1$ and $d = 5$, by considering periods as year, season of rice crop. By considering the background of IAAS, we fix the settings and details of these settings are presented in the Table 4.

Fixing Durations Based on Weather Forecasts by IMD. Based on the spatial and temporal scales of atmospheric systems and duration of the forecast IMD [1] defines various categories of forecasts. Currently, IMD is issuing various forecasts, the duration of each forecasts as explained below

- Now Casting: Current day Forecast with few hours forecast ahead
- Short Range Forecasts (SRF): Forecasts with duration of 1 day to 3 days.
- Medium Range Forecasts (MRF): Forecasts with duration 4 days to 10 days.
- Long Range Forecasts (LRF): Forecasts with duration 10 days to a season.

Based on weather forecasts provided by IMD, we have conducted experiments by considering CWCs for two durations. The duration $d = 1$ represents daily weather forecast and $d = 5$ represents average of weather data for five days. The expectation is that the experimental analysis of CWCs with $d = 1$ will provide the analysis of similar weather for the now casting and short range weather forecast and the experimental analysis of CWCs with $d = 5$ will provide the analysis for the medium range forecast.

Fixing the Categories of Weather Variables. Based on the influence of weather parameters on the domain, IMD has defined categories for specified range of each of these parameters which are called categories which are presented in Table 2. These categories are used to prepare weather summaries for forecast data. The category for the temperature variable is defined based on the value of historical normal. Climatologists define a historical normal as the arithmetic mean of a climate element for 30 year interval. Consider that the daily values of $\{RF, Tmax, Tmin, RHmax, RHmin\}$ for 30 years (1984–2014) is given. The climatic normal of each variable for the year 2015, is equal to the mean of the corresponding daily values of each weather variable $\{RF, Tmax, Tmin, RHmax, RHmin\}$ from the year 1984 to 2014 i.e., for 30 years. Climatologists understand the trends of both forecast and observed data using the corresponding historical normals. We use the notion of historical or climatic normal to decide the weather categories of temperature data.

Table 2. Category defined by India Meteorological Department [1]

Weather variable name	Range/deviation from normal	Description
Rain Fall (mm)	0–0	No Rain (NR)
	0.1–2.4	Very Light Rain (VLR)
	2.5–7.5	Light Rain (LR)
	7.6–35.5	Moderate Rain (MR)
	35.6–64.4	Rather Heavy Rain (RH)
	64.5–124.4	Heavy Rain (HR)
	124.4–244.4	Very Heavy Rain (VHR)
	>= 244.5	Extremely Heavy Rain (EHR)
Temperature (°C)	−1, 0, 1	Little Change (LC)
	2 or −2	Rise/Fall (R/F)
	3 to 4	Appreciable Rise (AR)
	−3 to −4	Appreciable Fall (AF)
	5 to 6	Marked Rise (MR)
	−5 to −6	Marked Fall (MF)
	>= 7	Large Rise (LR)
	<= −7	Large Fall (LF)
Relative Humidity (%)	0–30	Low (L)
	31–60	Moderate (M)
	61–80	High (H)
	>= 81	Very High (VH)

Fixing Periods: We have conducted the experiments by considering the period as a year and seasons of rice crop. The period of rice crop from the month of sowing to the month of harvesting with respect to kharif, rabi and summer seasons are presented in Table 3.

About the crop seasons: In Telangana, the agricultural seasons are defined as three main crop seasons based on the distinct weather situation on a longer term.

- Kharif: The duration is from June to October. The crops are sown at the beginning of the southwest monsoon and harvested at the end of the southwest monsoon.

- Rabi: The duration is from November to March. In this season crops need relatively cool climate during the period of growth, but warm climate during the germination of their seed and maturation.
- Summer: The duration is from March to May. Hot conditions prevail.

Table 3. Season-wise period of Rice crop from sowing to harvesting

Season name	Month of sowing	Month of harvesting
Kharif	June	October
Rabi	November	March
Summer	March	May

Table 4. Experiment settings

Periods of experiment	Year, Kharif, Rabi, Summer
Duration (d)	1 year, 5 years
Weather variables	$Tmax$, $Tmin$, $RHmax$, $RHmin$, RF
Categories of weather variables	Given in Table 2

4.3 Dataset Description

Dataset consists of 30 years of weather data i.e., from the year 1986–2015 of Rajendranagar region from Hyderabad, Telangana state in India. Each year consists of daily weather conditions of currently year i.e., 365 weather conditions for a non leap year and 366 weather conditions for leap year each with 5 parameters: RF, $Tmax$, $Tmin$, $RHmax$, and $RHmin$. The historical normals for each year were also collected.

We have conducted experiments by considering WCs for $d = 1$ and $d = 5$. The duration $d = 1$ represents the daily WC. We extract 365 WCs with $d = 1$, for each year. For $d = 5$, we get 73 WCs for a year. The value of RF is the cumulative value of corresponding daily RF values. The value of other parameter is equal to the mean of the corresponding daily values. From WCs for $d = 1$ and $d = 5$, we computed the corresponding coupled weather conditions and assigned domain specific categories (CWCs).

In total, we are considering the following variables: $Tmax$, $Tmin$, $RHmax$, $RHmin$, and RF. Given a domain, it may not be necessary that all the weather parameters play an equal role in decision-making process. For instance, in Agriculture based DSS we can ignore rainfall parameter for all the crops whose cultivation depends on the irrigation based sources like wells, bore wells etc. So, we have conducted the experiments by varying the number of weather variables in CWC from one to five that are as follows: 1-CWC, 2-CWC, 3-CWC, 4-CWC and 5-CWC. The number of CWCs combinations for each variable are presented in Table 5.

Table 5. Types of CWC and No. of combinations

Type of CWC	No. of combinations	List of combinations
$1-CWC$	$\binom{5}{1}=5$	$\langle\{Tmax\}\rangle$, $\langle\{Tmin\}\rangle$, $\langle\{RHmax\}\rangle$, $\langle\{RHmin\}\rangle$, $\langle\{RF\}\rangle$
$2-CWC$	$\binom{5}{2}=10$	$\langle\{Tmax, Tmin\}\rangle$, $\langle\{Tmax, RHmax\}\rangle$, $\langle\{Tmax, RHmin\}\rangle$, $\langle\{Tmax, RF\}\rangle$, $\langle\{Tmin, RHmax\}\rangle$, $\langle\{Tmin, RHmin\}\rangle$, $\langle\{Tmin, RF\}\rangle$, $\langle\{RHmax, RHmin\}\rangle$, $\langle\{RHmax, RF\}\rangle$, $\langle\{RHmin, RF\}\rangle$
$3-CWC$	$\binom{5}{3}=10$	$\langle\{Tmax, Tmin, RHmax\}\rangle$, $\langle\{Tmax, Tmin, RHmin\}\rangle$, $\langle\{Tmax, Tmin, RF\}\rangle$, $\langle\{Tmax, RHmax, RHmin\}\rangle$, $\langle\{Tmax, RHmax, RF\}\rangle$, $\langle\{Tmax, RHmin, RF\}\rangle$, $\langle\{Tmin, RHmin, RF\}\rangle$, $\langle\{Tmin, RHmax, RF\}\rangle$, $\langle\{Tmin, RHmin, RF\}\rangle$, $\langle\{RHmax, RHmin, RF\}\rangle$
$4-CWC$	$\binom{5}{4}=5$	$\langle\{Tmax, Tmin, RHmax, RHmin\}\rangle$, $\langle\{Tmax, Tmin, RHmax, RF\}\rangle$, $\langle\{Tmax, Tmin, RHmin, RF\}\rangle$, $\langle\{Tmax, RHmax, RHmin, RF\}\rangle$, $\langle\{Tmin, RHmax, RHmin, RF\}\rangle$
$5-CWC$	$\binom{5}{5}=1$	$\langle\{Tmax, Tmin, RHmax, RHmin, RF\}\rangle$

We have analyzed the similarity among the coupled weather conditions for the period of the year and seasons of rice crop for the following types of CWCs for durations $d=1$ and $d=5$.

- 1-CWC: The set of CWCs with one Variable.
- 2-CWC: The set of CWCs with two Variables.
- 3-CWC: The set of CWCs with three Variables.
- 4-CWC: The set of CWCs with four Variables.
- 5-CWC: The set of CWCs with five Variables.

4.4 Performance Metric

We define the performance metric to measure the extent of similar CWCs of given period with reference to preceding periods. That is, given a year n we would like to calculate the percentage of CWCs of period p for n^{th} year which are similar to CWCs of the periods of preceding $(n-1)$ years, where (n$\geq$2).

Definition 7. Coverage Percentage: Let CWC(p_i) be the set of category-based coupled weather conditions for a given period p_i of current year x. Let sum $(CWC(p_j))$ be the set of category-based coupled weather conditions defined over period p_j of preceding n years. $CP(p_i/n)$ is given by the percentage of CWCs of x which appear in the set of CWCs of the preceding n years. It is equal to $\frac{|CWC(p_i) \cap CWC(p_j)|}{|CWC(p_i)|}$.

4.5 Results

The results in Figs. 1 and 2 represents the consolidated results of all CWCs combinations represented in Table 5 with respect to 1-CWC, 2-CWC, 3-CWC, 4-CWC and 5-CWC, for the period of Year and Season with duration $d = 1$ and $d = 5$.

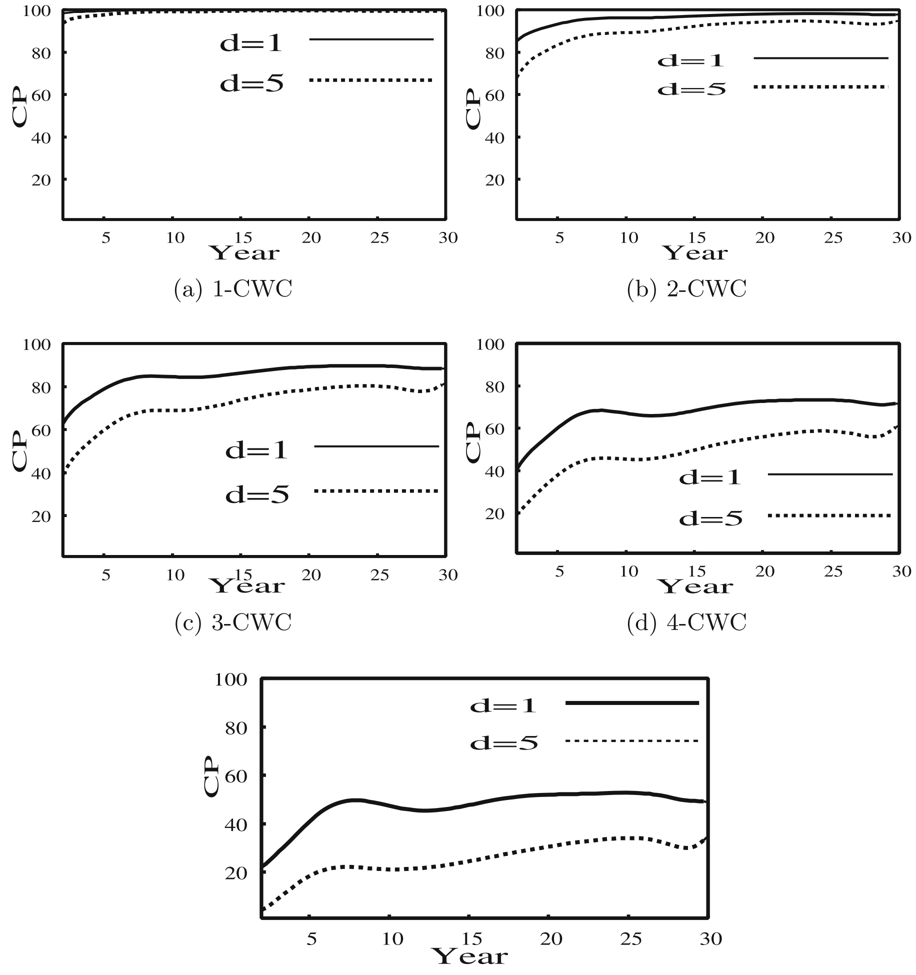
The experiments with 1-CWC analyze the coverage percentage of a given period for preceding $(n - 1)$ years CWCs to the period of current year (n^{th} year) CWCs with respect to one weather variable, similarly 2-CWC, 3-CWC, 4-CWC, 5-CWC analyze the coverage percentage of two, three, four and five weather variables respectively. In this experiment we have considered 5 weather parameters: RF, $Tmax$, $Tmin$, $RHmax$, and $RHmin$.

We represent year number on the x-axis and Coverage Percentage (CP) on the y-axis. The year number "1" represents 1986. For each year, we plotted coverage percentage of respective CWCs of that year similar to the CWCs of preceding years. For example, the CP(3/2) equal to the number of CWCs in 1988 which are similar to the CWCs set of 1986 and 1987. The coverage percentage represents the percentage of reuse with respect to each year for the given 30 years.

The results in Figs. 2 and 3 represent the mean CP values of the corresponding combinations 1-CWC, 2-CWC, 3-CWC, 4-CWC, and 5-CWC given in Table 5. The CP values of each combination of 1-CWC/2-CWC/3-CWC/4-CWC/5-CWC were found to be nearly the same. So, the corresponding mean CP values were plotted.

Experiments with *Period* = *Year*: In this experiment, we have analyzed the year-wise CP with respect to 1-CWC, 2-CWC, 3-CWC, 4-CWC and 5-CWC for duration $d = 1$, $d = 5$ considering the period as year. The results in Fig. 2 show that, the CP of 1-CWC, 2-CWC, 3-CWC, 4-CWC and 5-CWC has increased over years. Table 6 provides the summary statistics of Fig. 2. The results indicate that the CP values decrease as the number of variables in CWC increased. For example, for the given year the CP values for 1-CWC is higher than 5-CWC. However it can be observed that the value of 3-CWC is 73% even after three years. After seven years, the CP value is about 88%.

Experiments with *Period* = *Summer, Kharif* & *Rabi* : In this experiment, we have analyzed the CP by considering period as summer, kharif & rabi with respect to 1-CWC, 2-CWC, 3-CWC, 4-CWC and 5-CWC for duration $d = 1$



(a) 1-CWC

(b) 2-CWC

(c) 3-CWC

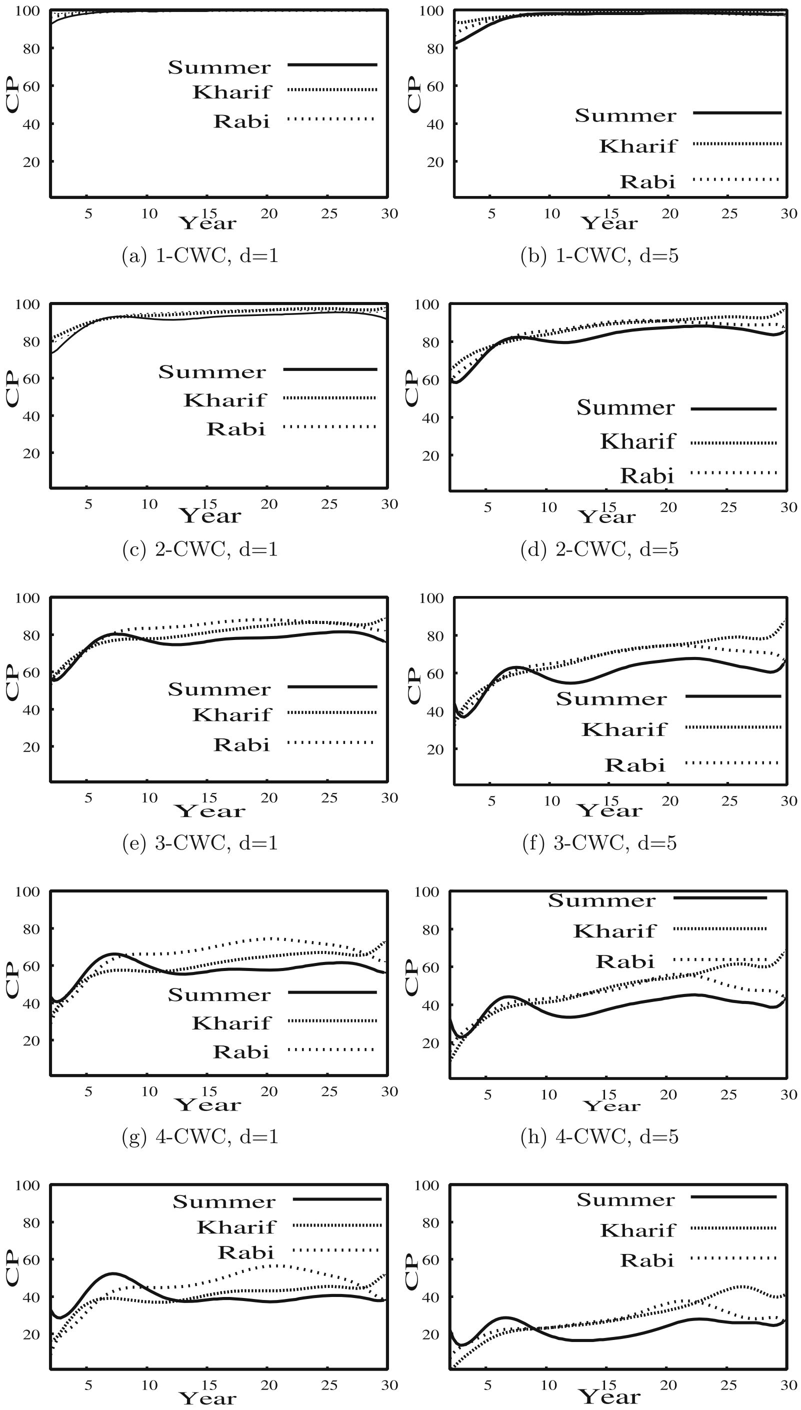
(d) 4-CWC

(e) 5-CWC

Fig. 2. CP with respect to $d = 1$ and $d = 5$ for $Period = Year$.

and $d = 5$. The results in Figs. 3a, c, e, g, i and b, d, f, h, j represents the CP of 1-CWC, 2-CWC, 3-CWC, 4-CWC and 5-CWC for $d = 1$ and $d = 5$ respectively.

The results represented in Fig. 3 show that, the CP of 1-CWC, 2-CWC, 3-CWC, 4-CWC and 5-CWC increased over years. Also, the statistics presented in Table 7 represents the average CP for period = rabi with respect to 1-CWC, 2-CWC, 3-CWC, 4-CWC and 5-CWC for third, fifth, seventh and tenth years. The results with respect to 1-CWC, 2-CWC and 3-CWC for $d = 1$ show that, the CP of third year (CP(3/2)) are greater than 97%, 86% and 69% respectively and the average CP of fifth-year i.e., (CP(5/4)) are greater than 99%, 91% and 75% respectively. Also, the results of seventh-year i.e., (CP(7/6)) are greater than



(a) 1-CWC, d=1

(b) 1-CWC, d=5

(c) 2-CWC, d=1

(d) 2-CWC, d=5

(e) 3-CWC, d=1

(f) 3-CWC, d=5

(g) 4-CWC, d=1

(h) 4-CWC, d=5

(i) 5-CWC, d=1

(j) 5-CWC, d=5

Fig. 3. CP with respect to d = 1 and d = 5 for *Period* = *Summer*, *Kharif* & *Rabi*.

99%, 95% and 86% respectively. We can observe an increase in reuse percentage of CWCs from (CP(3/2)) to (CP(7/6)). In the similar way we can find there is a significant reuse of 1-CWC, 2-CWC from (CP(3/2)) to (CP(5/4)) for $d = 5$. Though, the average CP values of 3-CWC for $d = 5$ were initially found to be slightly less when compared to average CP for $d = 1$, there is an increase in the average CP value over years.

Results Summary. The CP of 3, 5, 7 and 10 years for yearly, season-wise experiments for duration $d = 1$ and $d = 5$ are presented in Tables 6 and 7 respectively.

Table 6. CP Statistics of No. of (#) CWC for Period = Year

Duration	d = 1					d = 5				
Year/#-CWC	1-CWC	2-CWC	3-CWC	4-CWC	5-CWC	1-CWC	2-CWC	3-CWC	4-CWC	5-CWC
3	99.23	91.21	73.74	51.21	22.25	98.33	79.72	50.42	25	5.97
5	99.78	93.57	77.01	54.45	33.24	96.94	80.14	53.89	31.67	15
7	99.89	97.5	88.74	75	56.04	99.44	92.5	75.14	54.72	29.69
10	99.95	95.16	82.31	65.11	47.8	98.33	87.08	66.94	43.89	19.4

Table 7. CP Statistics of No. of (#) CWC for period = Rabi

Duration	d = 1					d = 5				
Year/#-CWC	1-CWC	2-CWC	3-CWC	4-CWC	5-CWC	1-CWC	2-CWC	3-CWC	4-CWC	5-CWC
3	97.73	86.87	69.47	48.67	31.33	94.48	70.34	47.24	33.1	12
5	99.33	91.47	75.6	54.67	30	96.55	75.52	54.48	40	28
7	99.6	95.53	86.47	70	52.67	98.62	87.24	70	40.59	22.33
10	99.87	97	87.2	72	48.67	99.31	89.66	70.34	42.69	25.67

The results show that the CP of 1-CWC, 2-CWC and 3-CWC have notable CP values that prove that there is a significant reuse of advisory prepared by considering only one, two and three weather variables. During the process of advisory preparation, the domain expert may consider only few weather variables that influence the weather by ignoring the rest of these variables. The above results show that there is a scope for significant reuse of the advisory based on fewer weather variables. We can observe high reuse by the end of third and fifth years for 1-CWC, 2-CWC and moderate reuse for 3-CWC. Apart from this, we can also observe that there is an increasing trend in the reuse of CWCs over 30 years for the periods of yearly and season-wise CWCs. The yearly experiments can be mapped to domain specific applications that rely on the day to day weather data. The season experiment maps to the weather based application to provide decisions for the agriculture domain.

5 Summary and Conclusion

Weather plays an important role in almost all aspects of the life. Hence, accurate and timely forecasting of weather helps in planning activities of many domains like increasing the agricultural production, providing precautionary measures to save life and property etc. Based on the climatic conditions, the crop development in every country is divided into various seasons. The impact of weather on each of this stage may be different as “Climatic requirement of each phase may be different and also each phase is sensitive to the certain environmental parameters as well as management practices”. In this work, we have analyzed the similarity among coupled weather conditions for five weather variables. The coupled weather conditions represent the observed and forecasted weather values of given duration.

In the proposed framework, we have analyzed the similarity among the CWCs for the period of the year and seasons over the years by exploiting the domain specific categories. We computed the similarity with respect to individual weather variables as DSSs of agriculture domain may not depend on all weather parameters for decision making. We have varied the number of variables in weather condition with one day duration and five days duration and computed coverage percentage over 30 years of real weather data by considering categories defined by India Meteorological Department.

The results show that the CP of CWCs with three weather variables for the period of year is about 77% after five years. Also, the CP of CWCs with two variables is about 97% and three variables is about 88% after seven years. The results are very positive. The results indicate that significant degree of similarity exists among 1-CWC, 2-CWC and 3-CWC over the years for daily and five day CWCs. The results provide the scope to develop automatic weather-based DSS in various domains with minimal human intervention and improving the utilization of the generated content. Overall, the results of the above framework prove that if we build a DSSs which depends on both observed and forecasted weather for advice preparation there is a very high reuse of the advisory, provided we have domain specific category values for each weather variable.

As part of future work, we would like to analyze the changing patterns in the weather over a period of time for a given domain for duration *d*. The notion of weather condition will be improved by capturing the change between the observed weather and the weather forecast. We planned to extend the framework to other domains to compute the coverage percentage of weather conditions by applying domain specific categories of weather variables such as health, crops, livestock and so on.

Acknowledgement. This work is supported by India-Japan Joint Research Laboratory Project entitled “Data Science based farming support system for sustainable crop production under climatic change (DSFS)”, funded by Department of Science and Technology, India (DST) and Japan Science and Technology Agency (JST).

References

1. India meteorological department weather forecasters guide. http://imdpune.gov.in/Weather/Reports/forecaster_guide.pdf
2. Bajwa, S.: Investigating the impact of weather information on travel time prediction. In: OPTTIMUM International Symposium on Recent Advances in Transport Modelling (2013)
3. Balasubramanian, T., Jagannathan, R., Maragatham, N., Sathyamoorthi, K., Nagarajan, R.: Generation of weather windows to develop agro advisories for Tamil Nadu under automated weather forecast system. J. Agrometeorology **16**(1), 60 (2014)
4. Filip, F.G.: Decision support and control for large scale complex systems. In: Large Scale Complex Systems Theory and Applications, vol. 11, pp. 2–12 (2007)
5. Johnson, M.P., Zheng, K., Padman, R.: Modeling the longitudinality of user acceptance of technology with an evidence-adaptive clinical decision support system. Decis. Support Syst. **57**, 444–453 (2014)
6. Jones, J.W., Hansen, J.W., Royce, F.S., Messina, C.D.: Potential benefits of climate forecasting to agriculture. Agric. Ecosyst. Environ. **82**(1), 169–184 (2000)
7. Maini, P., Rathore, L.: Economic impact assessment of the agrometeorological advisory service of India. Current Sci. **101**(10), 1296–1310 (2011)
8. Mamatha, A., Krishna Reddy, P., Kumara Swamy, M., Sreenivas, G., Reddy, D.R.: A framework to improve reuse in weather-based decision support systems. In: Srinivasa, S., Mehta, S. (eds.) BDA 2014. LNCS, vol. 8883, pp. 1–13. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13820-6_1
9. Monkhouse, F.J.: A Dictionary of Geography. Transaction Publishers, New Brunswick (2007)
10. Rathore, L.: Weather information for sustainable agriculture in India. J. Agric. Phys. **13**(2), 89–105 (2013)
11. Reddy, P.K., Trinath, A.V., Kumaraswamy, M., Reddy, B.B., Nagarani, K., Reddy, D.R., Sreenivas, G., Murthy, K.D., Rathore, L.S., Singh, K.K., Chattopadhyay, N.: Development of eagromet prototype to improve the performance of integrated agromet advisory service. In: Madaan, A., Kikuchi, S., Bhalla, S. (eds.) DNIS 2014. LNCS, vol. 8381, pp. 168–188. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05693-7_11
12. Krishna Reddy, P., Bhaskar Reddy, B., Rama Rao, D.: A model of virtual crop labs as a cloud computing application for enhancing practical agricultural education. In: Srinivasa, S., Bhatnagar, V. (eds.) BDA 2012. LNCS, vol. 7678, pp. 62–76. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35542-4_6
13. Schepers, H., Bouma, E., Frahm, J., Volk, T., Secher, B.: Control of fungal diseases in winter wheat with appropriate dose rates and weather-based decision support systems 1. EPPO Bull. **26**(3–4), 623–630 (1996)
14. Selby, R.W.: Enabling reuse-based software development of large-scale systems. IEEE Trans. Softw. Eng. **31**(6), 495–510 (2005)
15. Srivastava, B.: A decision-support framework for component reuse and maintenance in software project management, pp. 125–134. IEEE (2004)
16. Stanescu, I.A., Filip, F.G.: Capture and reuse of knowledge in ICT-based decisional environments. Informatica Economica **13**(4), 11 (2009)
17. Strahler, A.N.: Introduction to physical geography (1965)

Holistic Analysis of Multi-source, Multi-feature Data: Modeling and Computation Challenges

Abhishek Santra[1](✉) and Sanjukta Bhowmick[2]

[1] Information Technology Laboratory, CSE Department,
University of Texas at Arlington, Arlington, TX, USA
abhishek.santra@mavs.uta.edu

[2] Department of Computer Science, University of Nebraska at Omaha,
Omaha, NE, USA

Abstract. As a result of our increased ability to collect data from different sources, many real-world datasets are increasingly becoming multi-featured and these features can also be of different types. Examples of such multi-feature data include different modes of interactions among people (Facebook, Twitter, LinkedIn, ...) or traffic accidents associated with diverse factors (speed, light conditions, weather, ...).

Efficiently modeling and analyzing these complex datasets to obtain actionable knowledge presents several challenges. Traditional approaches, such as using single layer networks (or monoplexes) may not be sufficient or appropriate for modeling and computation scalability. Recently, multiplexes have been proposed for the elegant handling of such data.

In this position paper, we elaborate on different types of multiplexes (homogeneous, heterogeneous and hybrid) for modeling different types of data. The benefits of this modeling in terms of ease, understanding, and usage are highlighted. However, this model brings with it a new set of challenges for its analysis. The bulk of the paper discusses these challenges and the advantages of using this approach. With the right tools, both computation and storage can be reduced in addition to accommodating scalability.

Keywords: Big data analytics · Multi-source, disparate data
Multiplex · Graph analysis and query processing
Lossless composability · Aggregation functions

1 Introduction

Data analytics requires a suite of various techniques to analyze different kinds of datasets and derive meaningful conclusions from them. Holistic analysis relates to analyzing a multi-feature dataset by including the effect of different combinations of features or perspectives. In this paper, we discuss a network-based model that is suited for a large class of problems. We present the utility of this model and its concomitant computing challenges.

As an example, consider the problem of modeling and analyzing the traffic accident problem or data set for a region or a country. A number of features are

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 59–68, 2017.
https://doi.org/10.1007/978-3-319-72413-3_4

associated (and collected) with each accident such as location, speed, time of the day, severity of the accident, light, weather, and road conditions. One may want to analyze this dataset from multiple angles: general accident prone regions, dominant feature associated with most accidents, ordering features based on their effect on the severity of the accident, effect of individual or combination of features on accidents in a region or across all regions.

Consider another dataset where we have information about scientists who collaborate with each other, cities that have direct flights, and conferences that have overlapping research topics. In addition, there is information about who lives in which city and the cities in which annual conferences have been held. Given this dataset, it would be useful to understand: whether a large group of collaborators have attended several conferences, which group of conferences have the largest number of papers from a group of collaborators, which is the best city to hold a workshop on a particular topic to get maximum number of collaborating scientists.

Note that, unlike the earlier problem where the features referred to the same entity set (accidents), in this problem different features are captured for different disjoint entity sets (scientists, cities, and conferences). The analysis may span multiple entities and their relationships in different ways.

Traditionally, *graphs* (which we also term as monoplexes) are used for representing and analyzing systems of interacting entities [18,21]. Typically, entities are represented as vertices. Two vertices are connected by a single edge, which represents a common value of the feature between the two entities. This representation can be extended by introducing multiple edges between vertices for each different type of feature. Instead of using multiple edges which make the representation as well as analysis of graphs difficult, we propose to use multiplexes (multiple layers of interconnected graphs) as an alternative model.

In this paper, we elaborate on the benefits of using different types of multilayer networks (homogeneous, heterogeneous, and hybrid multiplexes) for modeling and associated computation challenges for doing holistic analysis. In contrast to the vast amount of work on analyzing monoplex networks, the research on multiplexes is considerably sparse. Even when the systems are modeled as multilayer networks, they are studied only for *very specific problems in a subdiscipline* [6,20].

We provide a brief overview of work related to multi-feature data analysis in Sect. 2. We will discuss modeling benefits and issues in Sect. 3 and the computational issues in Sect. 4. In Sect. 5, we give an overview of our preliminary work that addresses some of the challenges highlighted in this paper. We will end with the conclusions in Sect. 6.

2 Related Work

Recently, many analytical tasks have used multilayer networks for partitioning the space of *well-defined explicit interactions* among the *same entity set* [7,15,16,22,25,27]. Most of the work have tried to figure out overall multiplex diagnostics such as degrees and distances by considering the multiplex layers individually or all of them together. In contrast, we focus on different types of features and entity sets and efficient analysis of arbitrary combinations of multiple layers.

Tensor Representations have also been used for certain multi-feature data representation [13]. They are mainly used for *node-aligned networks*, that is networks having same set of nodes. We are dealing with networks that are both node-aligned (homogeneous multiplex) and not node-aligned (heterogeneous multiplexes).

Graph mining (e.g., substructure discovery [14,17,23], AGM [4], FSG [14], or pattern-growth - gSpan [33], FFSM [19] and GASTON [28], disk-based approaches [5,30] and SQL-based approaches [10,29]) has been researched extensively as compared to graph querying [12]. To the best of our knowledge, graph mining and querying techniques have not been much studied for multiplexes.

3 Modeling Using Multiplexes

Multi-feature data comprises of multiple relations existing among the same or different types of entities. Relationships among the entities can either be specified by explicit interactions (like flights, co-authors and friends) or based on a similarity metric depending on the type of the feature like nominal, numeric, time, date, latitude-longitude values, text, audio, video or image.

For each feature, monoplexes will represent the relationship through directed/undirected (denoting information flow) and weighted/unweighted (quantifying relationship strength) edges between the entities, denoted by nodes. However, such monoplexes have to be generated for every feature or combination of features by repeatedly scanning the datasets and evaluating the similarity metrics. Another alternative is to use multiple edges between nodes corresponding to the features they are related to. But, in this model, for any k-feature based analysis, the entire graph will have to be loaded and traversed in order to first extract the set of desired edges. Further, such a convoluted representation makes the visualization process tedious.

In order to address the drawbacks of monoplex-based modeling, in this paper we propose the use of multiplexes, a form of *network of networks*. In this case, every layer represents a distinct relationship among entities with respect to a single feature. The sets of entities across layers, which may or may not be of the same type, can be related to each other too. Formally, a multiplex is defined by a set of n graphs $G_1(V_1, E_1)$, $G_2(V_2, E_2)$, ..., $G_n(V_n, E_n)$ and a set of edges $E_{1|2}, E_{2|3}, \ldots, E_{n-1|n}$. Each graph G_i is formed of the vertex set V_i, and the intra-layer edge set E_i. The inter-layer edge set $E_{i|j}$, connects the vertices of G_i to the vertices of G_j. Therefore, in contrast to monoplexes, for holistic analysis, the pre-processing cost is significantly reduced as the desired individual layers are either readily available or multi-feature composed layers can be generated by combining the edges of the individual layers through cost-effective set operations. Development of efficient lossless techniques for combining k individual layers translating to a new composed perspective is challenging due to the variation in edge connectivity, edge weight domain and edge directions in each layer.

Based on the type of relationships and entities, multiplexes can be of different types. Layers of a **homogeneous multiplex** are used to model the diverse relationships that exist among the **same type of entities** like traffic accidents (Fig. 1(a)). Therefore, $V_1 = V_2 = \ldots = V_n$ and inter-layer edge sets are empty as no relations across layers are necessary. Relationships among **different types of entities** like cities (connected by flights), scientists (connected to collaborators) and conferences (related by overlapping research domains) are modeled through **heterogeneous multiplex** (Fig. 1(b)). The inter-layer edges represent the relationship across layers like conference venues, scientist residences and conference attendance. In addition to being collaborators, scientists may be friends on Facebook or connected on ResearchGate or LinkedIn. Thus, for modeling multi-feature data that capture **multiple relationships within and across different types of entity sets**, a combination of homogeneous and heterogeneous multiplexes can be used, called **hybrid multiplexes**.

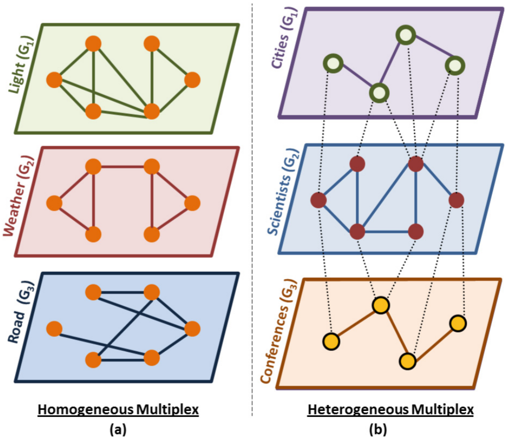


Fig. 1. Basic types of multiplexes

Benefits of Multiplex-based Modeling: Modeling of multi-feature data as multiplexes allows *ease of handling the dataset incrementally* through the addition of nodes (when a new accident, scientist, city or conference is encountered), edges (to represent the new entity's relationships with the earlier entities) or layers (to account for fresh perspectives). Moreover, a *latest snapshot* of multiplex can be easily maintained through the deletion of obsolete entities (nodes), relationships (edges) or perspectives (layers). Further, this modeling provides a medium to **study the relationships among the entities with respect to individual or combinations of features or perspectives**.

4 Multi-feature Computations Using Multiplexes

The major task is to be able to perform computations on the multi-feature data for holistic understanding. A plethora of algorithms are available for analyzing monoplex-based models. However, the limitations highlighted in modeling multi-feature data as a monoplex makes this medium unfavorable. On the other hand, the amount of work done for efficiently analyzing different types of multiplexes is at a nascent stage. For instance, there is hardly any work pertaining to mining and querying of multiplexes.

The traditional computational techniques proposed for monoplexes can be leveraged to perform analysis of multiplexes with respect to any combination of features (or layers). However, for holistic understanding of multi-feature data with a multiplex with n layers, $2^n - 1$ layer combinations need to be analyzed. Thus the major issue is the exponential increase in the overall computational costs with respect to both time and storage space in the presence of large number of layers ([7] has used 300 layers). This challenge highlights that the need of the hour is the **development of robust algorithms that are able to compute network characteristics, mine interesting hidden patterns and query different combinations of multiplex layers in a cost effective manner**. Additional challenges to perform specific computations on the two basic types of multiplexes have been discussed in the following sections.

4.1 Homogeneous Multiplex Computations

For the traffic accident scenario, the effectiveness of accident prevention measures and the dominance of factors can be studied through the variation in the accident-prone regions over time. In graph terminology, it translates to finding out groups of tightly connected vertices called communities (through random walks [8], maximizing modularity [26] or maximimizing permanence [9]). Therefore, for such computations we need to **devise efficient techniques for generating communities with respect to any combination of multiplex layers**. Similarly, it will be beneficial to **develop methods to compute the relative ordering and correlation among different feature (or layer) combinations based on their importance**. For example, if road conditions have *more impact* on accidents than light, then more funds can be allocated to fix the roads as compared to lights. An added challenge in this regard will be to **identify metrics that can quantify the importance of a layer** based on semantics of the domain. Density, number of influential nodes (high closeness and high betweenness centrality vertices), core-periphery structure and local and global clustering coefficients are few alternatives for such a metric.

Any of the above techniques should be efficient enough to be able to reduce the exponential complexity of generating, storing and analyzing every layer combination. **Formulation of efficient aggregation functions that can combine the results from n individual layers to compute the results of any layer combination** is a way forward. The **layer-wise analysis results will be in diverse formats** like substructures (communities), real numbers (density, clustering coefficients) or sets (hubs, high centrality nodes, nodes in inner core), adding to the complexity of this challenge. Further, the **performance of different types of network structures** for the formulated functions needs to be understood using evaluation metrics like NMI, Purity, ARI and Jaccard Index [24]. Moreover, it should be noted that there may be a class of computations for which the result of the combined layer cannot be re-constructed from the layer-wise results. For such cases, **obtaining a confidence interval for aggregation functions** will be useful to approximate results of the layer combinations.

4.2 Heterogeneous Multiplex Computations

In single networks (monoplexes) important vertices have been defined with respect to information flow through high degree, betweenness and closeness centrality vertices. However, in the case of heterogeneous multiplexes apart from the intra-layer connectivity, the inter-layer connectivity also needs to be considered. Therefore, in the city-scientist multiplex (extracted from Fig. 1(b)), important cities will be the ones that are not only easily accessible but also where most sought after collaborators reside (marked in red in Fig. 2). Thus, the challenge in this case is to **devise efficient ways to compute high centrality vertices across multiple connected layers**. It should be noted that **a high centrality vertex in one layer, may not also be a high centrality vertex in the combined layer**.

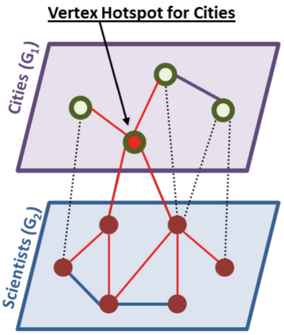


Fig. 2. Vertex hotspot for cities with respect to scientists (Color figure online)

In heterogeneous multiplexes, the formulation of aggregation functions that combine the layer-wise results becomes more challenging as the **results of the bipartite graph formed by the inter-layer edges** also have to be taken into account. One must consider that **the layers may be connected not just sequentially one after another** (i.e. layer A connected to layer B connected to layer C) but can be **connected in different directions** (i.e. layers A, B and C can be connected to each other in a triangle).

In monoplex networks, mining of interesting substructures of different sizes using metrics like Minimum Description Length (MDL) [17] or frequency is a well-explored field. However, to **develop algorithms for mining on multiplexes** the notion of **subgraphs and patterns** in a multiplex needs to be articulated for using a metric like *MDL* and the **anti-monotonic property of metrics** like *frequency* has to be established. The city-scientist multiplex in Fig. 3 with scientist node labels depicting research fields, illustrates an example of a *frequent pattern in a multiplex*. Another challenge will be **defining exact and similar (or inexact) substructures**. Further, **strategies to partition a multiplex need to be devised** for extending the existing scalable mining techniques based on graph partitioning and map/reduce [11].

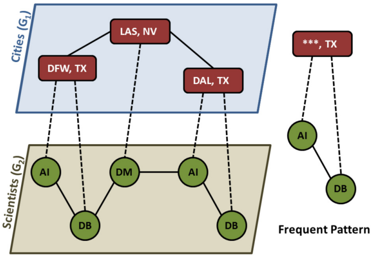


Fig. 3. Example of frequent pattern in a multiplex

Querying is for verifying the existence of known patterns or extracting all instances of partially specified patterns. Queries can be of different types, for example - cities where scientists attending *most number of conferences* reside (node degree based), cities where scientists belonging to *largest group of collaborators* reside (community based), best possible city where a well-connected group of collaborators can meet up by taking the *minimum number of flights* (path based) etc. For such type of analysis, **query processing algorithms for queries on multiplexes have to be developed**. Few challenges in coming up with these algorithms are - determining the order (in parallel or as a partial order) to process layers for efficiency, generating metric to evaluate alternate query plans, evaluating the suitability of an index-based or substructure expansion-based approach and identifying query processing requirements in terms of the graph properties.

5 Preliminary Work

In this section, we will provide an overview of our preliminary work that addresses some of the challenges highlighted in this paper.

In [31], we have proposed the combination of undirected and unweighted homogeneous multiplex layers using the Boolean operators - AND, OR, NOT. For example, the AND-composed layer consists of only those edges (or relationships) that are present in all the constituent individual layers. This work proposes an intersection based aggregation method that just uses the layer-wise communities to accurately re-create the communities of any AND-composed multiplex layer, provided the communities of the individual layers are self-preserving in nature. We have shown empirically using real-life multi-feature datasets (traffic accidents [2] and storms [3]) that this community re-creation process leads to an overall saving of over 40% in computation time. Currently, we are extending this AND re-creation process to handle any type of layer-wise communities. Moreover, we are also addressing the various challenges like merging or splitting of communities based on the extent of their overlap across layers in order to formulate the community re-construction method for OR-composed multiplex layers. Metrics like modified normalized mutual information (modified-NMI) [24] that consider network topology are being used for evaluating the quality of the re-constructed communities.

Apart from communities, another recent work of ours [32] concentrates on efficiently estimating the central (or influential) entities or hubs across AND-composed homogeneous multiplex layers by using the layer-wise centrality results. Variation in the edge connectivity across individual layers can cause non-hubs to become hubs and hubs to become non-hubs in the AND-composed layers, thus making the hub estimation process a non-trivial task. Here we have developed various efficient heuristics based on degree and closeness centrality metrics by maintaining minimal neighborhood information from the individual layers. Experiments on diverse real-life multi-feature datasets (traffic accidents [2] and IMDb [1]) have shown that the proposed heuristics estimate more than 70–80% of the central vertices while reducing the overall computational time by at least 30%. Currently, we are in the process of generalizing and extending this work to other centrality measures like betweenness and eigenvector and combination methods involving disjunction (OR) and negation (NOT).

6 Conclusions

In this position paper, we have discussed the relevance of multiplexes for modeling multi-feature data as well as the computational advantages. Holistically analyzing multi-feature data can benefit from a representation that is easy to understand, visualize, and at the same time has advantages from a computation perspective.

The computational challenges identified in this paper are being addressed by us [31,32] and the larger research community. Solutions to these challenges will enrich the data analytics repertoire making it easier to analyze problems that can benefit from graph-based representation.

Acknowledgment. We would like to extend our gratitude towards Dr. Sharma Chakravarthy, University of Texas whose insight and expertise greatly helped in shaping up of this position paper.

References

1. The internet movie database. ftp://ftp.fu-berlin.de/pub/misc/movies/database/
2. Road safety - accidents (2014). https://data.gov.uk/dataset/road-accidents-safety-data/resource/1ae84544-6b06-425d-ad62-c85716a80022
3. Storm events database by NOAA. https://www.ncdc.noaa.gov/stormevents/ftp.jsp
4. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Very Large Data Bases, pp. 487–499 (1994)
5. Alexaki, S., Christophides, V., Karvounarakis, G., Plexousakis, D.: On storing voluminous RDF descriptions: the case of web portal catalogs. In: International Workshop on the Web and Databases, pp. 43–48 (2001)
6. Berenstein, A., Magarinos, M.P., Chernomoretz, A., Aguero, F.: A multilayer network approach for guiding drug repositioning in neglected diseases. PLOS (2016)
7. Boden, B., Gnnemann, S., Hoffmann, H., Seidl, T.: Mining coherent subgraphs in multi-layer graphs with edge labels. In: Proceedings of the 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD 2012), Beijing, China, pp. 1258–1266 (2012)
8. Bohlin, L., Edler, D., Lancichinei, A., Rosvall, M.: Community detection and visualization of networks with the map equation framework (2014). http://www.mapequation.org/assets/publications/mapequationtutorial.pdf
9. Chakraborty, T., Srinivasan, S., Ganguly, N., Mukherjee, A., Bhowmick, S.: Permanence and community structure in complex networks (2015). Accepted to TKDD
10. Chakravarthy, S., Pradhan, S.: DB-FSG: an SQL-based approach for frequent subgraph mining. In: DEXA, pp. 684–692 (2008)
11. Das, S., Chakravarthy, S.: Partition and conquer: map/reduce way of substructure discovery. In: Madria, S., Hara, T. (eds.) DaWaK 2015. LNCS, vol. 9263, pp. 365–378. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22729-0_28
12. Das, S., Goyal, A., Chakravarthy, S.: Plan before you execute: a cost-based query optimizer for attributed graph databases. In: Madria, S., Hara, T. (eds.) DaWaK 2016. LNCS, vol. 9829, pp. 314–328. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-43946-4_21

13. De Domenico, M., Solé-Ribalta, A., Cozzo, E., Kivelä, M., Moreno, Y., Porter, M.A., Gómez, S., Arenas, A.: Mathematical formulation of multilayer networks. Phys. Rev. X **3**(4), 041022 (2013)
14. Deshpande, M., Kuramochi, M., Karypis, G.: Frequent sub-structure-based approaches for classifying chemical compounds. In: IEEE International Conference on Data Mining, pp. 35–42 (2003)
15. Domenico, M.D., Nicosia, V., Arenas, A., Latora, V.: Layer aggregation and reducibility of multilayer interconnected networks. CoRR abs/1405.0425 (2014). http://arxiv.org/abs/1405.0425
16. Dong, X., Frossard, P., Vandergheynst, P., Nefedov, N.: Clustering with multi-layer graphs: a spectral perspective. CoRR abs/1106.2233 (2011). http://dblp.uni-trier.de/db/journals/corr/corr1106.html#abs-1106-2233
17. Holder, L.B., Cook, D.J., Djoko, S.: Substucture discovery in the SUBDUE System. In: Knowledge Discovery and Data Mining, pp. 169–180 (1994)
18. Horvath, S., Zhang, B., Carlson, M., Lu, K., Zhu, S., Felciano, R., Laurance, M., Zhao, W., Qi, S., Chen, Z., et al.: Analysis of oncogenic signaling networks in glioblastoma identifies ASPM as a molecular target. Proc. Nat. Acad. Sci. **103**(46), 17402–17407 (2006)
19. Huan, J., Wang, W., Prins, J.: Efficient mining of frequent subgraphs in the presence of isomorphism. In: ICDM 2003, Washington, DC, USA, pp. 549–552 (2003)
20. Huang, C.Y., Wen, T.H.: A multilayer epidemic simulation framework integrating geographic information system with traveling networks. In: 2010 8th World Congress on Intelligent Control and Automation (WCICA), pp. 2002–2007, July 2010
21. Jeong, H., Mason, S.P., Barabási, A.L., Oltvai, Z.N.: Lethality and centrality in protein networks. Nature **411**(6833), 41–42 (2001)
22. Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J.P., Moreno, Y., Porter, M.A.: Multilayer networks. CoRR abs/1309.7233 (2013). http://arxiv.org/abs/1309.7233
23. Kuramochi, M., Karypis, G.: Frequent subgraph discovery. In: IEEE International Conference on Data Mining, pp. 313–320 (2001)
24. Labatut, V.: Generalized measures for the evaluation of community detection methods. CoRR abs/1303.5441 (2013)
25. Magnani, M., Rossi, L.: Formation of multiple networks. In: Greenberg, A.M., Kennedy, W.G., Bos, N.D. (eds.) SBP 2013. LNCS, vol. 7812, pp. 257–264. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37210-0_28
26. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**, 026113 (2004)
27. Ng, M.K.P., Li, X., Ye, Y.: Multirank: co-ranking for objects and relations in multi-relational data. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1217–1225. ACM (2011)
28. Nijssen, S., Kok, J.N.: A quickstart in frequent structure mining can make a difference. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 647–652, KDD 2004. ACM, New York (2004)
29. Padmanabhan, S., Chakravarthy, S.: HDB-Subdue: a scalable approach to graph mining. In: DaWaK, pp. 325–338 (2009)
30. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C.: PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth. In: ICDE, pp. 215–224 (2001)

31. Santra, A., Bhowmick, S., Chakravarthy, S.: Efficient community re-creation in multilayer networks using boolean operations. In: International Conference on Computational Science, ICCS 2017, 12–14 June 2017, Zurich, Switzerland, pp. 58–67 (2017). https://doi.org/10.1016/j.procs.2017.05.246
32. Santra, A., Bhowmick, S., Chakravarthy, S.: Hubify: efficient estimation of central entities across multiplex layer compositions. In: 2017 IEEE International Conference on Data Mining Workshops, ICDM Workshops 2017, New Orleans, USA, 18 November 2017 (2017, to appear)
33. Yan, X., Han, J.: gSpan: graph-based substructure pattern mining. In: IEEE International Conference on Data Mining, pp. 721–724 (2002)

Information and Knowledge Management

Entity Markup for Knowledge Base Population

Lili Jiang(✉)

Department of Computing Science, Umeå University, Umeå, Sweden
lili.jiang@cs.umu.se

Abstract. Entities (e.g. people, places, products) exist in various heterogeneous sources, such as Wikipedia, web page, and social media. Entity markup, like entity extraction, coreference resolution, and entity disambiguation, is the essential means for adding semantic value to unstructured web contents and this way enabling the linkage between unstructured and structured data and knowledge collections. A major challenge in this endeavor lies in the ambiguity of the digital contents, with context-dependent semantic and dynamic. In this paper, I introduce the main challenges of coreference resolution and named entity disambiguation. Especially, I propose practical strategies to improve entity markup. Furthermore, experimental studies are conducted to fulfill named entity disambiguation in combination with the optimized entity extraction and coreference resolution. The main goal of this paper is to analyze the significant challenges of entity markup and present insights on the proposed entity markup framework for knowledge base population. The preliminary experimental results prove the significance of improving entity markup.

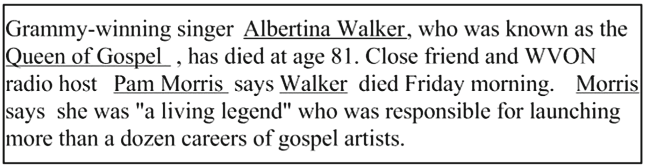
1 Introduction

Entity markup, like entity extraction, coreference resolution, and named entity disambiguation, is the essential means to deliver semantic value to unstructured web contents and enable the linkage between unstructured and structured data and knowledge bases. Named entity disambiguation (NED) is a task of linking mention in given text to a unique entity in existing knowledge base (i.e. Wikipedia). NED is one of many importation operations for data management, information retrieval, semantic mining. Further research in entity disambiguation is necessary to help promote information quality and improve data reporting in multidisciplinary fields requiring accurate data representation.

Despite many advances in the last few years, fully automatic NED is inherently difficult and may also be computationally expensive [6,16,17,20,26,36]. NED methods have been shown to perform very well for prominent entities mentioned in high-quality texts like news articles, but they degrade in terms of both precision and recall when dealing with lesser known long-tail entities. Since advanced methods utilize machine learning or extensive statistics for semantic relatedness measures among entities, the availability of labeled training data is usually a big bottleneck.

However, even if we had perfect NED methods for aligning ambiguous names in text documents with canonicalized entities registered in a knowledge base, the

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 71–89, 2017.
https://doi.org/10.1007/978-3-319-72413-3_5



Grammy-winning singer Albertina Walker, who was known as the Queen of Gospel , has died at age 81. Close friend and WVON radio host Pam Morris says Walker died Friday morning. Morris says she was "a living legend" who was responsible for launching more than a dozen careers of gospel artists.

Fig. 1. A text illustrating mentions in NED

envisioned cross-linkage between unstructured web contents and semantic data collections would still have big gaps. For example, considering the text snippet in Fig. 1, both “Queen of Gospel” and “Walker” refer to “Alberta Walker”, but automated algorithms typically lack the background knowledge is challenging. Another example in the text of Fig. 1 is that “Pam Morris” is relatively unambiguous but the isolated mention “Morris” in NED is more challenging. The reason is that finding the correct link in NED requires disambiguating based on the mention string and often non-local contextual features. However, we can nevertheless capture their mentions under different names and try to gather equivalence classes of text phrases that refer to the same entity. This is known as the problem of coreference resolution(CR) [13,34,35,37]. The other reason of the failed NED is the dynamic world: new entities come into existence. When facing such emerging entities, CR methods are also helpful. In addition to dealing with the recognized and emerging entities, CR methods can also help to increase the recall of NED for known entities, simply by capturing more surface phrases (e.g. [23,28]). For example, we should discover mentions such as “Donald Trump” and “the USA president” to infer that they denote the same entity. We can map more text mentions onto entities, thus improving NED recall at high precision. Systematically gathering different mentions names for entities is the problem of Dictionary Building. It has been studied in the literature, harnessing href anchor texts, click logs, and other assets [18,39]. However, doing this for emerging entities that are not yet registered in a knowledge base is a largely unexplored task.

This paper presents a framework for entity markup, where I combine entity extraction, coreference resolution, and named entity disambiguation in a joint manner. During this process, practical strategies are proposed to get optimum results.

2 Terminology, Problem, and Framework

2.1 Terminologies

- **Entity:** Any object existing in the real world can be entity, such as person, organization, location, and product.
- **Mention:** Mention is the surface name, which an entity is referred in text. In other words, mention is the instance of entity. For example,

"Albertina Walker" can be the mention of entity "Albertina Walker" (en.wikipedia.org/wiki/Albertina_Walker).

- **Entity Extraction** (EE)**:** The input text (e.g., web pages, news articles, etc.) is processed to discover *mentions* of named entities, that is, surface phrases that are likely to denote individual entities (as opposed to common noun phrases). Our implementation currently uses the Stanford NER Tagger [10] (a trained CRF) and Illinois Mention Detector [2] for this purpose.
- **Name Entity Disambiguation** (NED) is the process of linking the named mentions in text to entities registered in the existing knowledge bases (e.g., Wikipedia). Mention "Albertina Walker" could be easily linked to the American gospel singer Albertina Walker in Wikipedia. However, the following "Walker" may refer to numerous distinct candidates: "Alice Walker", "Derek Walker", or "Kara Walker". Mention "Pam Morris" should be linked to *Null* as it has no corresponding RDF triples in Knowledge base.
- **Entity Candidate:** Possible entities (with unique canonical names) from Knowledge base, a mention may denote. We harness existing knowledge bases like DBpedia or YAGO.
- **Coreference Resolution** (CR) is the process of finding all the mentions (i.e. named mention, nominal mention, and pronoun mention) in documents that refer to the same entity. Taking the given example text, mentions "singer Albertina Walker", "Albertina Walker", "Queen of Gospel" as well as "Walker" refer to the same entity[1].
- **Coreference Equivalence Class:** Coreference equivalence class (aka., coreference chain) is the set of all the mentions, which refer to the same entity in a given text. For example, here we can have two coreference equivalence classes {singer Albertina Walker, Albertina Walker, Queen of Gospel, Walker, she} and {Pam Morris, Morris}.

2.2 Problem Formulation

Given a document d, we extract all mentions and formulate as $M = \{m_1, m_2, \ldots, m_{n_m}\}$, where all mentions are linearly ordered by positions. We define entity candidate list as $E = \{E_1, E_2, \ldots E_{n_m}\}$, where $E_i = \{e_{i1}, e_{i2}, \ldots\}(0 < i \leq n_m)$ denotes the entity candidate list of mention m_i.

After that, we propose a coreference resolution classifier to generate coreference equivalence classes $C = \{C_1, C_2, \ldots C_{n_c}\}$, where C_i denotes a single coreference equivalence class and $C_i \cap C_j = \emptyset$ $(1 \leq i, j \leq n_c)$. $coreferent(m_i, m_j)$ is true if mentions m_i and m_j are within the same coreference equivalence class. The core task of our work is as follows: given the mentions $M = \{m_1, m_2, \ldots, m_{n_m}\}$ extracted from document d, we first generate coreference equivalence classes $C = \{C_1, C_2, \ldots, C_{n_c}\}$ and entity candidates $E = \{E_1, E_2, \ldots E_{n_m}\}$. Based on M, C, and E, we built a mention-entity graph as shown in Fig. 2. Let $\psi(m_i, e_{ij})$ be a score function reflecting the likelihood that candidate entity e_{ij} is the correct disambiguation linking entity for $m_i \in M$. Let $\phi(m_i, m_j)$ be a score function

[1] http://en.wikipedia.org/wiki/Albertina_Walker.

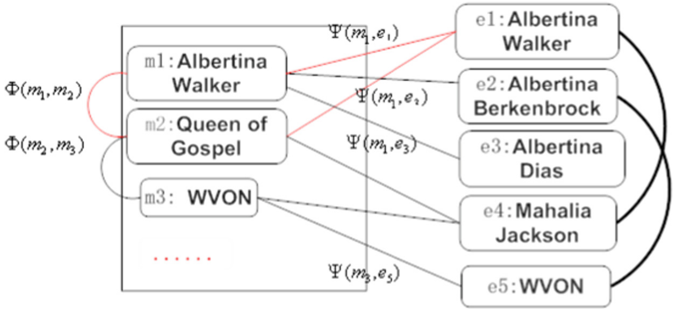


Fig. 2. A mention-entity graph example in NED

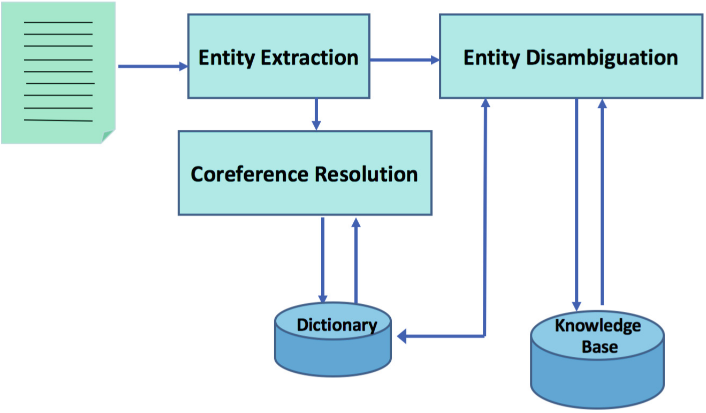


Fig. 3. An overall framework for entity markup

reflecting the likelihood that mention m_i and mention m_j are *coreferent*. Let $\kappa(e_i, e_j)$ be a score function reflecting the coherence of entity e_i and entity e_j. Scanning from m_1 to m_{n_m} using random walk with restart, ψ, ϕ, and κ work together to link each mention on the left side to a unique entity on the right side, and we finally generate the entity candidate list as $E' = \{E'_1, E'_2, \ldots E'_{n_e}\}$, where each element is a ranked list of entities $\{e_{i1}, e_{i2}, \ldots\}$ registered in Knowledge base $\mathcal{KB}$ for each named mention $m_i \in M$. Meanwhile, we output updated results for coreference equivalence classes $C' = \{C'_1, C'_2, \ldots C'_{n_{c'}}\}$.

2.3 Entity Markup Framework Overview

Figure 3 gives a pictorial overview of the proposed framework of entity markup. It consists of three main functional components: entity extraction (EE), followed by coreference resolution (CR) and entity disambiguation (NED). Details will be explained in the following sections.

3 Entity Extraction and Sieve

Entity extraction is also called entity recognition or mention extraction. We recognize all the mentions using the state-of-the-art NER models from Stanford [10] and Illinois [2].

After recognition, we first filter out some nonsense or incorrectly extracted entity mentions. Secondly, we remove or correct some nominal mentions by exploring the position relations (i.e. overlapping and embedding) between named mentions and nominal mentions.

- For any mention, we will filter it out if it meets any condition as follows: (1) it is consist of stop words; (2) it contains too many punctuation; (3) if it is started with conjunction or ended with a conjunction (e.g. as) or pronoun (e.g. his, her); (4) it contains incomplete punctuation (i.e. half bracket or quotation mark); (5) a mention is one word and the word prior to or following this mention is a noun phrase.
- For embedding mentions, we keep both of them (e.g., “Dutch” embedded in “Dutch Soccer Captain”). For overlapping mentions, four false positive cases will be considered: (1) only one of them is recognized correctly, such as “Sen. Bill” and “Bill Frist”; (2) both of them are recognized correctly, such as “Irishman Patrick Butler” and “16-year-old Irishman”; (3) neither of them are recognized correctly but an integrated them is a correct noun phrase. For example, “President Ali” and “Ali Abdullah Saleh” could be integrated as President Ali Abdullah Saleh; (4) both mentions are recognized and the only difference is definite article, such as “The Justice” and “Justice”. In the following, we identify all the overlapping mentions, and then filter out them or modify them to correct form. For an embedding pair of named mention m_i and nominal mention m_j, if m_j contains certain stop words/determiners (e.g. ‘a’, ‘an’, ‘the’, ‘something’, ‘anything’, ‘nothing’, ‘there’, or ‘here’) as prefixes or suffixes, and m_i equals m_j after removing these articles, we only keep m_i. If the predicted mention type of m_i is *PERSON*, *ORGANIZATION*, or *LOCATION*, and both m_i and m_j are consist of nouns, we will merge m_i and m_j into a single nominal mention. For any other embedding pairs, we will drop m_j from mention list M. After the sieve, an updated mention list M is obtained for further use in the following sections.

4 Coreference Resolution

An effective coreference resolution system is an important component in any NLP pipeline that deals with language understanding tasks, such as question answering and information extraction. In this paper, we regard it critical for named entity disambiguation task. We first provide detailed error analysis with examples regarding the different kinds of errors that show up in the basic coreference resolution system (Sect. 4.1). According to these error analysis, we propose additional features and constrains to obtain coreference equivalence classes for a given document (Sect. 4.2).

4.1 Error and Challenge Analysis

We test the basic coreference system (i.e. Illinois CR system) on the news corpus and analyze the results. Errors decreasing precision and recall are categorized with examples according to their causes.

Lack of Alias Detection. Missing aliases (e.g. nickname, acronyms) reduces the performance of coreference resolution. The following three examples show three pairs of missing aliases in the state-of-the-art CR systems.

> Richard N. Gottfried is a leading policy-maker nationally... Dick Gottfried is also a member of the Steering Committee...

> But now some residents are worried that the Gansevoort Peninsula, also known as Pier 52, will continue to be used ...

> The Delaware Department of Transportation (DelDOT) is an agency of the U.S....

Inaccuracy of Appositive. Three main appositive errors are observed.

Geographical location mismatch. Some false positives are caused due to the mismatch between cities and countries. Taking the following as example, as a borough in New Jersey, “Fair Haven” is incorrectly resolved to be coreferent with NJ.

> POWER-Thomas C., of Fair Haven, NJ, died at home on October 8, 1997.

Mismatch of preposition and head word. In the adverbial modifier with a preposition, the head word is usually incorrectly resolved to be *coreferent* with the subject after it (e.g. “South Florida” and “a weary public” as follows).

> Around South Florida, a weary public was trying to cope with fears...

Entity mismatch within an entity set. Entities belonging to the same entity set are sometimes resolved as appositive falsely, when they are displayed in a row with comma as separators.

> Aluvial slopes are inhabited by Pedunculate Oak, linden, European hornbeam, and European Turkey oak.

Overall, according to our experimental results, the appositive for people formed as (*proper noun, common noun*) are returned with high precision and low recall. The appositive formed as (*proper noun, proper noun*) is usually determined incorrectly.

Side-effects from string similarity. String similarity is essential and yet risky in coreference resolution. As shown in the example as follows, "Mr. Clinton" and "Hillary Rodham Clinton" are incorrectly resolved to be *coreferent*.

Mr. Clinton was accompanied by his wife, Hillary R. Clinton.

4.2 Coreference Resolution Learning and Inference

In previous section, we analyzed some errors from the state of the arts in coreference resolution. As coreference resolution data is not totally linearly separable, in this case, learning with further inference outperforms either local classifier or global classifier when the number of training examples is not sufficiently large [31]. In this section, we proposed a two-stage method for coreference resolution. Firstly in the learning stage, we train a local classifier for each pair of mentions, and generate a score indicating pairwise probability of coreference resolution. Herein, each mention pair suffices symmetry. Secondly in the inference stage, we employ deterministic constrains to aggregate the scores generated by the classifier, and then link mention pairs into coreference equivalence classes. Herein, we fix the transitivity volition. Finally, the coreference equivalence classes formed by only one mention will be deleted.

Learning. We train a logistic regression classifier with a probability $\phi(m_i, m_j)$ as output for each pair of mentions m_i and m_j. It refers to the probability that m_i and m_j is *coreferent*. After learning, coreference resolution score of pairwise mention is a function $M \times M \rightarrow [0, 1]$, where 0 and 1 are the minimum and maximum coreference score.

$$\phi(m_i, m_j) = \frac{1}{1 + \sum_k \exp(w_k * f_k(m_i, m_j))} \tag{1}$$

Where w_k is the weight vector learned from training data, f is the feature vector and $f_k(m_i, m_j)$ is the value of the kth feature.

We create training samples according to the widely used method from [38]. Given a mention m_j from training data, this method generates positive samples with m_j and its closest preceding coreferent mention m_i, and negative samples with m_j and every intervening mention $m_{i+1}\ m_{i+2}\ \ldots\ m_{j-1}$.

Learning Features. Table 1 provides a concise view for all the features we used in the learning phase. Details about these features are explained in the following.

Co-occurring distribution probability. We use a model based on knowledge base $\mathcal{KB}$ (YAGO) to get a prior probability that two mentions linking to the same entity. In $\mathcal{KB}$, we have anchor link for each mention to a Wikipedia entry page (i.e. entity). Thus, the occurrence frequency of each mention and the frequency of its linking to an entity are obtained. The probability $p(m_i, e)$ ($e \in E(m_i)$, $m_i \in M$) is defined as the fraction between the number of occurrences of m_i

Table 1. Learning features

Feature Type	Features	Description
Popularity	Popularity(m_i,m_j)	The probability that mentions denote the same entity
People coreference	Person(m_i,m_j)	The probability that mentions denote the same person
	isSameGender(m_i, m_j)	True if the person mentions has the same gender, and False otherwise
Alias	PatternAlias(m_i,m_j)	True if the mentions is a pattern-based alias of the other, and False otherwise
	KBAlias(m_i,m_j)	True if the mentions is a knowledge-based alias of the other, and False otherwise
	Acronym(m_i,m_j)	The probability that a mention is an acronym of the other
	Abbreviation(m_i,m_j)	The probability that a mention is an abbreviation of the other
Relation	isInRelation(m_i,m_j)	True if there is relation word (e.g. wife, husband) between these two mentions, and False otherwise
	isNounInPreposition(m_i,m_j)	True if one mention is in the a preposition phase, followed by the other mention, and False otherwise
	isLocationHerachy(m_i,m_j)	True if these two mentions are in the different level of a location hierarchy tree (e.g. m_i is a state, while m_j is a country)
String match	SubString(m_i,m_j)	True if one of the two mentions is the substring of the other, False otherwise
	Head(m_i,m_j)	True if these two mentions has the same head word, False otherwise
	Jaccard(m_i,m_j)	Jaccard measure
	DF(m_i,m_j)	TFIDF measure
Distance	CharacterDistance(m_i,m_j)	Normalized distance between two mentions according to characters
	WordDistance(m_i,m_j)	Normalized distance between two mentions according to words
	SentenceDistance(m_i,m_j)	Normalized distance between two mentions according to sentences
Entity type	IsPerson(m_i)	True if m_i is predicted as PERSON, and False otherwise
	IsORGANIZATION(m_i)	True if m_i is predicted as ORGANIZATION, and False otherwise
	IsMISC(m_i)	True if m_i is predicted as unknown, and False otherwise
	TypeMatch1(m_i, m_j)	True if predicted entity types are identical but not unknown, and False otherwise
Mention type	IsNominalMention(m_i)	True if m_i is a nominal mention, and False otherwise

in $\mathcal{KB}$ actually referring to e, and the total number of occurrences of m_i in $\mathcal{KB}$ as mention. Assume the probability for each mention is independent, the probability that two mentions m_i, m_j denote the same entity can be calculated as $p(m_i, m_j) = p(m_i, e)(m_j, e)$.

People-oriented resolution. Two definitions are given firstly: (1) half name: person name with only one token or an appellation plus a single token (e.g. Jack, Mary, Mr. Smith); (2) full name: person name (exclusive appellation words) with token size larger than one (e.g., John Smith, George W. Bush).

It is found that a rather high percentage of documents contain person names, so we specially propose an algorithm as feature for person name coreference resolution, based on the following observations: (1) if the full name of a person is mentioned explicitly at least once in a document, the corresponding half name is usually used to refer to the same person in its context; (2) for full name, it may be referred by different half names, for example, "Richard Abruzzo" was mentioned as "Richard" as well as "Abruzzo" in the same document; (3) for half name, the literally same half name may denote different full names, for example "Bob" denotes both "Bob Behn" as well as "Robert D. Behn" in the same document; (4) for a half name, its full name is usually found right ahead of it at least once, especially when the half name is mentioned the first time.

Given a document, we extract all person names, and divide them into a full name list and a half name list. For each pair of (*full name, half name*), we compute a score about how likely they denote the same person as output. This score is computed based on string similarity p_{ssim}, lexicon-based nickname similarity p_{lsim}, and positional similarity p_{psim}. For example, "Richard" and "Richard Stallman" has a string similarity as 0.5. In our lexicon, "Bob" and "Robert" has a probability of 0.9 to denote the same person. For each pair of half name and full name, its positional similarity depends on the number of full names between them and ahead them. We compute the score using the following formula:

$$\begin{aligned} p(h,f) &= (p_{ssim}(h,f) + \rho p_{lsim}(h,f)) \\ &\quad \times (1 + \theta p_{psim}(h,f)) \\ \rho &= \begin{cases} 1 & (p_{ssim}(h,f) = 0) \\ 0 & (p_{ssim}(h,f) > 0) \end{cases} \end{aligned} \tag{2}$$

Where h is a half name, and f is a full name. $p_{ssim}(h,f)$ is their string-based similarity in terms of Jaccard ratio. $p_{lsim}(h,f) = p_{ssim}(h_n,f) * p_{occur}(h_n,h)$ is the lexicon-based similarity according to person nickname lexicon. h_n is h's nickname extracted from lexicon, for example, "Dick" and "Richard" are nickname with each other. $p_{occur}(h_n,h)$ is the probability that h_n is likely to be used to represent h. $p_{psim}(h,f)$ is the positional similarity between h and f, where $p_{psim}(h,f) = (N(f_h) - N(f_b) - N(f_a))/N(f_h)$, and $N(f_b)$ is the number of full names between h and f, $N(f_a)$ is the number of full names ahead of both h and f, while $N(f_h)$ is the number of full names containing h. Note that only full name with string similarity or lexical similarity will be considered in positional similarity. There are two factors in Eq. 2: ρ is used to active lexical similarity when string similarity equals zero. θ is 1 if f appears ahead of h, otherwise, θ is 0.5. A running example is described as follows, h ("Dick") and f ("Richard Stallman") appear in the same document. $p_{ssim}(h,f)$ equals 0, ρ equals with 1. For h_n ("Richard"), their $p_{lsim}(h,f)$ is computed as $0.5 * 0.85 = 0.425$, $p_{psim}(h,f)$ is assumed to be 0.6, thus the final $p(h,f)$ is $0.425 * (1 + 0.6) = 0.68$.

After that, each pair of half name and full name in given document is assigned a score bounded in [0,1]. Note that the proposed person coreference resolution does not consider the match between full names. For example, if

Table 2. Nickname patterns

Pattern1	Pattern2	Pattern3
aka	*aka*	*known as*
better known as	*nee*	*nickname of*
alias	*whose real name is*	*nickname for*
also known as	*was born*	
nickname	*is/was/once called*	
is/was/once called		

"George W. Bush", "George W. H. Bush" and "Bush" appear in the same document. It will find "Bush" to match one of them, and never explore whether these two full names represent the same person, which will be handled in the named entity disambiguation component in Sect. 5.

Alias detection. We detect aliases based on pattern and knowledge base respectively. (1) pattern: Table 2 shows three types of alias patterns through extending the patterns in [3], "`mention` *pattern1* `alias`", "`alias` *pattern2* `mention`", and "*pattern3* `mention alias`". (2) knowledge base: we query mentions against Freebase to get the alias attributes (e.g. common.topic.alias of Freebase). For example, "The Big Apple" and "The Melting Pot" are obtained by querying "New York City".

Acronym detection. Acronym is a special case for coreference resolution. For example, "Delaware Department of Transportation" may be mentioned by using its acronym "DelDOT". Regarding the special characteristics for detecting acronym, the naive patterns of "*expanded form (acronym)*" and "*acronym (expanded form)*" are very useful. In combination with these two naive patterns above and other two functions (i.e. $AcronymOnline(m_i)$ and $AcronymRules(m_i)$), we propose an algorithm to identify acronyms in coreference resolution as shown in Algorithm 1.

Algorithm 1. Acronym Detection

```
Input: $M$
Output: $AcronymMap$
for $m \in M$ do
    IsA = $IsAcronymGuo(m_i)$;
    if IsA=true then
        $AcronymMap \leftarrow AcronymOnline(m_i)$
    else
        $AcronymMap \leftarrow AcronymRules(m_i)$
```

$AcronymMap$ is a hashmap with mention as key and an acronym list as value. For each mention $m_i \in M$, we first judge whether m_i is an acronym of some other mention using an effective function $IsAcronymGuo$ [12] (line 2). This function recognizes mention m_i as an acronym, if and only if mention m_i satisfies the following conditions: (1) it contains no more than 4 letters with no less than 2 upper case letters; (2) it must not contain more than 2 lower case letters. If m_i is acronym, we further search all its acronym expansion using online acronym detector[2] $AcronymOnline(m_i)$ given m_i as query. After that we only keep mentions of $AcronymOnline(m_i)$, which exist in the given document (line 3–4). And then we add all pairs (m_i, m_j) in $AcronymOnline(m_i)$ to $AcronymMap$. If m_i is not an acronym, we generate acronym for m_i based on hand-crafted rules, including constructing its acronym by getting the initial capital letters of m_i, extracting the patterns of "*expanded form (acronym)*" or "*acronym(expanded form)*" (line 5–6). Then we extract all other mentions $m_j \in M$ meeting the patterns with m_i to form $AcronymRules(m_i)$. After that, we add all detected pairs (m_i, m_j) in $AcronymRules(m_i)$ to $AcronymMap$.

Relation detection (boolean). For any two mentions m_i and m_j, we detect the following three boolean features: (1) relation detection: if relation cues exist between mentions m_i and m_j (i.e. wife, husband, aunt, uncle, nephew and etc.). (2) preposition detection: if m_i is the head mention in adverbial modifier following a preposition, and m_j is the subject in the modified sentence. (3) location mismatch detection: if both m_i and m_j are locations and belong to different levels in a location hierarchy. For example, in the previous mentioned example, "Galway" is city, while "Ireland" is a country. These features are motivated by the observation that some mentions are most unlikely to be *coreferent*, if some special relation (e.g. above three relations) exists between them.

String match (boolean, double). For any two mentions m_i and m_j, we get the following three features according to their surface string. (1) if m_i is the substring of m_j. (2) if m_i and m_j have the same head word, such as "Grammy-winning singer Albertina Walker" and "Albertina Walker". (3) string similarity: following the stop-words removal (e.g. a, an, the, of, and), we use two basic state-of-the-art measures, Jaccard and TFIDF to obtain a string similarity between m_i and m_j.

Distance measure. We use three types of distance features, which respectively count how many characters, words, and sentences apart the two given mentions are. These features are motivated from our observations that for different types of *coreferent* mentions, their distance features are not always the same. For example, abbreviation mentions are usually laid closely, while appositive and pattern-based aliases are often in the same sentence. Acronym coreference mentions may be laid closely or apart from each other by sentences.

Entity type and mention type (boolean). We use existing natural language processing tool (i.e. Stanford NER) to predict the entity type of each mention, and also check whether a pair of mentions are identical in terms of PERSON, ORGANIZATION, or LOCATION. Moreover, mention type of nominal mention

[2] http://acronyms.silmaril.ie/.

is considered as a feature. These features are motivated from the fact that entities of different types (i.e. PERSON, ORGANIZATION, and LOCATION) have different characteristics, some further processing can be used to handle each of them specially.

Inference. Pairwise classifier is simple and flexible with successful achievements in previous research studies. However, it has disadvantage that it is possible that these independent decision will not be consistent with each other (i.e. transitivity violation). For example, mention m_i and m_j are deemed *coreferent*, m_h and m_j as *coreferent*, there is no guarantee that the classifier will deem m_i and m_h as *coreferent*. After pairwise classifier, we have to do inference to ensure the transitivity consistence: when mentions *coreferent*(m_i, m_j) and *coreferent*(m_j, m_k) are true, *coreferent*(m_i, m_k) must be true.

We propose the following constrains in inference phase based on error analysis introduced in Sect. 4.1. Constraints are used to enforce accurate coreference resolution at testing time [31]. For any mentions m_i and m_j, they should not be *coreferent* if they meet any one of the following four constrains: (1) *gender disagreement*: we detect person gender through extracting appellation words (e.g. ‘Mr.’, ‘Mrs.’, and ‘Miss’). For all full names and last names, we also use the US census to further predict the gender of the person name. (2) *number disagreement*. If either mention has numbers (e.g. product model number) and they are distinguishing digitals. (3) *category disagreement*. If both mentions are recognized in different categories with the same entity type (e.g. city and province). (4) *relation agreement*. If the coreference mentions are close with each other in position, and there is also a relation word between them.

Some rules above are overlapped with training features. However, there is no conflicts as some of them may be weakened in the training model and we strengthen them here. We built coreference equivalence classes through merging mention pairs in a consistent way, which meets the transitivity and constrains above.

5 Named Entity Disambiguation

There existed some works on named entity disambiguation [16,26,36], we first provide detailed error analysis with examples regarding the different kinds of errors that show up in the basic NED methods (Sect. 5.1). According to these error analysis, we propose a general random walk based solution for NED (Sect. 5.2).

5.1 Error and Challenge Analysis

According to the observations on results from the state-of-the-art methods of NED, some errors decreasing performance of named entity disambiguation are presented as follows.

Obsession over Prominent Entity. The state-of-the-art methods do well when the mentions are linked to prominent entities, this biases to a poor performance when they are working on long tail entities or more ambiguous mention. Taking the following text as example, "Albertina Walker" can be easily disambiguated as the American gospel singer, "Pam Morris" is disambiguated as *Null*. However, the following "Morris" is linked to the popular baseball player "Matt Morris" incorrectly. "Chicago" could be correctly linked to the US city, however it will be much more challenging if it denotes other non-prominent entities (e.g. basketball team, bank name).

> Close friend and WVON radio host Pam Morris says Albertina Walker died Friday morning in Chicago .... Morris says Walker was "a living legend" ....

Ambivalence on String Similarity. Undoubtedly, exact string match is effective in NED. The state-of-the-art methods mentioned above links "Don Evans" in the following example to "Don Evans" (`./wiki/Don_Evans`) or *Null* instead of the correct person "Donald Evans"(`./wiki/Donald_Evans`). The problem is that the correct one is sometimes exclusive from the high-ranking entity candidates in the prior stage based on the initial string similarity filtering.

> George W. Bush also named Don Evans as Secretary of Commerce.

Haste on Emerging Entities. All the emerging entities, which are not registered in existing knowledge bases are always linked to *Null* individually by most of the entity disambiguation methods. However, quite few of these methods explore the relevance between these emerging entities. For instance, all the underlined mentions as follows should be linked to *Null*, among which "Golden Managers Acceptance Corporation" and "Golden MAC" denote the same organization.

> Duff Co. downgraded the program of Golden Managers Acceptance Corporation from Duff 1+ to Duff 1. The assets in the Golden MAC program continue to ..

Classifying them into a coreference equivalence class will be beneficial to the knowledge base population for further use. In this example, a new entry page or disambiguation page could be created for them in Wikipedia or YAGO Knowledge base, and these two mentions in text should be redirected to the same entry page.

5.2 Random Walk Based Named Entity Disambiguation

Some errors in entity disambiguation are caused due to lack of coreference information (e.g. "Pam Morris" and "Morris"), which leads to low ranking or exclusive of the correct entity in the entity candidate list, while the biggest challenge in coreference resolution is lack of background semantic knowledge (e.g. “Albertina Walker” and “Queen of Gospel”). These two tasks should not be treated individually and it is ideal to correct prior errors from both tasks as more information is obtained in the following steps. Thus, we propose a robust graph based framework for NED.

With the entity extraction (EE), coreference resolution (CR), and named entity disambiguation (NED) from the previous steps, we build a mention-entity graph G as shown in Fig. 2. The left column contains the mentions $M = \{m_1, m_2, \ldots m_{n_m}\}$ extracted from given document, and we get an initial coreference resolution score $\phi(m_i, m_j)$ as edge weight for each pair of mentions using Eq. 1. Mentions within the same equivalence class are linked by solid edge, and other edges between mentions are marked using dashed lines. The right column contains the entity candidates $E = \{E_1, E_2, \ldots\}$ from Yago Knowledge base. We harness the existing knowledge bases (i.e. YAGO), which provides a catalog of entities and their surface names. AIDA [16] presents a disambiguation framework combining local context measurement and global coherence. (1) Local context measurement ψ_l. On the mention side, it collects all the tokens in given text as context. On the entity side, it considers the keyphrases or salient words, precomputed from Wikipedia articles. In addition, it uses WordNet to do syntactic contextualization to obtain phrases typically used with the same verb that appears with the mention in the input text. (2) Global coherence ψ_g. It qualifies the coherence between two entities by the number of incoming links in Wikipedia articles, This motivates from the fact that most texts deal with a single or a few semantically related topics such as rock music or internet technology. We use the similarity values ψ_l and ψ_g for a mention m and entity e respectively.

In Fig. 2, the initial $\phi(m_i, m_j)$ and $\psi(m_i, e)$ have been assigned by our coreference resolution classifier and AIDA disambiguation framework. We update the disambiguation edge weight $\psi(m_i, e)$ through combining the following functions of mention m in given document.

We used the random walk with restart probability α, i.e., the probability with which the random walk jumps back to seed node s, and thus “restarts”. Random walk models the distribution of rank, given that the distance random walkers can travel from their source (i.e., mention) is determined by alpha. At each step of random walk with a restart probability α, it jumps to a random node, and with probability $1 - \alpha$ follows a random outgoing edge from the current node. In fact the expected walk-length is $1/\alpha$. The formula now becomes $x' = (1-\alpha)Ax + \alpha E$. Here, alpha is the restart probability, which is a constant between 0 and 1, and E is the vector containing the source of information - i.e. in our case it is all zero, except for the red vertex where our information starts to spread. Ax is the node weight of mention m in the previous iteration, here E is obtained using the following formula, where α is fixed to 0.5 to keep the random walkers not to travel too far.

$$\begin{aligned} nodeweight(v)_{t+1} = {} & (1 - \alpha) \times nodeweight(v)_t \\ & + \alpha \sum edgeweight(v, w) * nodeweight(w) \end{aligned} \tag{3}$$

5.3 Dictionary Building and Knowledge Population

We handle two cases in dictionary building $dict(M, E, C)$ for mention m as follows: (1) Add linkable mention m to M; (2) The non-linkable mention are supposed to be the new emerging entities. Regarding these newly discovered entity, if there are several mentions within an equivalence class, a representative mention will be created and initial popularity value will be created. With the growing number of discovered coreferent mentions, its popularity value will be updated. When the popularity value is sufficient large, the newly discovered entity could be added to knowledge base. This part is worthwhile further exploration in future. Machine learning and Crowdsourcing techniques could be involved for screening and evaluating newly entities.

6 Experimental Study

6.1 Dataset

We used the following two public datasets for evaluation: (1) APW: 150 Associated Press news articles published on October 1st and 150 published on November 1st, 2010, taken from the GigaWord 5 corpus [32]. Mentions were extracted and matched to entities in Wikipedia as ground truth. (2) CONLL [27]: CoNLL 2003 data, which consists of proper noun annotations for 1393 Reuters newswire articles. All these proper nouns were hand-annotated with corresponding entities in YAGO2.

6.2 Evaluation and Discussion

As shown in Table 3, we use document precision, precision and MAP as evaluation measures for named entity disambiguation [16]. To quantify how the various aspects of our proposed strategies affect the performance of named entity disambiguation, we studied two variations. (1) baseline named entity disambiguation algorithm with random walk; (2) baseline with coreference resolution. (3) baseline with coreference resolution and optimized entity extraction. Experimental results shows the effectiveness of the optimized coreference resolution and entity extraction for entity disambiguation. In this paper, we aim to run through the whole entity markup framework, even with preliminary experimental results. More experimental studies could be conducted, and more advanced methods should be designed for a holistically optimized solution in entity markup.

According to the experimental results, future direction on entity extraction is still promising although it has been studied for many years. Quite a number of experimental errors are raised due to the *Geography ambiguity* especially in the United States. It is common for two cities(towns) sharing the same name in different states, such as “Burlington, New Jersey” and “Burlington, Vermont”. It is also common for the same name denoting both state and city, such as “New York” or “Washington”. State abbreviation is popular such as “Connecticut” and “Conn”. A gazetteer (toponymical dictionary), which is a geospatial dictionary of place names, must be beneficial here.

Table 3. Entity disambiguation evaluation

Data set	APW2010			CONLL-Test		
	Doc. Precision	Precision	MAP	Doc. Precision	Precision	MAP
Baseline	0.8163	0.8093	0.8076	0.7923	0.7424	0.7871
Baseline + CR	0.8168	0.81	0.809	0.8102	0.7587	0.7469
Baseline + CR + EE	0.8187	0.8144	0.834	0.8189	0.7707	0.8016

7 Related Work

Coreference resolution (CR) finds the mentions in text that refer to the same entity [13,19,35,37]. Entity coreference resolution is a well studied problem with many methods and tools [1,2,5,8,9,21,22,33,38,41]. CoNLL (the Conference on Natural Language Learning) 2011 [30] and 2012 [29] included a shared task of coreference resolution in which training and test data is provided by the organizers which allows participating systems to be evaluated and compared in a systematic way. Recently, more work showed that joint models resolve mentions across multiple entities result in better performance than simply resolving mentions in a pair-wise comparison. [22] introduces a joint coreference resolution model which combines events and entities by incorporating verbs from event as context features. [14] focuses on enhancing coreference resolution with named entity disambiguation in natural language processing tasks.

Named entity disambiguation (NED) [11] links the mentions in document to entities registered in the existing knowledge bases (e.g., Wikipedia). Earlier work [4,24] on entity disambiguation exploits local features (e.g., bag of words, n-grams), and compares the lexical context around the ambiguous mention to the content of the candidate disambiguation's Wikipedia text. Later on, extended resources are used to explore semantic features, and the most widely used resources includes WordNet [25], Freebase (www.freebase.com), and Yago [40]. Wikipedia also offers some helpful features, like redirection page, disambiguation page, infoboxe, category hierarchy, and hyperlink. Based on these, work on entity disambiguation has stressed on global features exploration [7,15,16,26], these approaches give high confidence to entity candidates, which are strongly related to each other within one document. Entity disambiguation systems with only local features are strong baseline hard to beat, and the systems combining both local and global features could get marginal improvements. However, the biggest challenge is to find tradeoff between local and global features as they have significant strengths and weaknesses of each [36]. Recent years, some work explored other natural language processing tasks to boost entity disambiguation, such as word sense disambiguation, relation extraction, and coreference resolution.

8 Conclusion

This paper introduces the importance and challenges in entity markup (i.e., entity extraction, coreference resolution, and named entity disambiguation).

A practical entity markup framework is proposed to enhance named disambiguation in combination with optimized entity extraction and coreference resolution. The running examples and preliminary experimental studies prove the proposed strategies to enhance entity markup, enriching knowledge base population.

Acknowledgment. Many thanks Johannes Hoffart and Gerhard Weikum for some discussions relevant with this paper. Thanks Stephan Seufert for his original version of random walk codes.

References

1. Aktolga, E., Cartright, M.A., Allan, J.: Cross-document cross-lingual coreference retrieval. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 1359–1360. CIKM (2008)
2. Bengtson, E., Roth, D.: Understanding the value of features for coreference resolution. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, pp. 294–303 (2008)
3. Bollegala, D., Honma, T., Matsuo, Y., Ishizuka, M.: Mining for personal name aliases on the web. In: Proceedings of the 17th international conference on World Wide Web, pp. 1107–1108, WWW 2008 (2008)
4. Bunescu, R.: Using encyclopedic knowledge for named entity disambiguation. In: EACL, pp. 9–16 (2006)
5. Chang, K.W., Samdani, R., Rozovskaya, A., Rizzolo, N., Sammons, M., Roth, D.: Illinois-coref: the UI system in the CONLL-2012 shared task. In: CoNLL Shared Task (2012)
6. Cornolti, M., Ferragina, P., Ciaramita, M.: A framework for benchmarking entity-annotation systems. In: Proceedings of the International Conference on World Wide Web (WWW), pp. 249–260 (2013)
7. Cucerzan, S.: Large-scale named entity disambiguation based on Wikipedia data. In: Proceedings 2007 Joint Conference on EMNLP and CNLL, pp. 708–716 (2007)
8. Durrett, G., Klein, D.: Easy victories and uphill battles in coreference resolution. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (2013)
9. Finin, T., Syed, Z., Mayfield, J., McNamee, P., Piatko, C.: Using wikitology for cross-document entity coreference resolution. In: Proceedings of the AAAI Spring Symposium on Learning by Reading and Learning to Read. AAAI Press (2009)
10. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proceedings of the Association for Computational Linguistics, ACL 2005 (2005). http://nlp.stanford.edu/software/CRF-NER.shtml
11. Getoor, L., Machanavajjhala, A.: Entity resolution: theory, practice & open challenges. Proc. VLDB Endow. **5**(12), 2018–2019 (2012)
12. Guo, Y., Qin, B., Li, Y., Liu, T., Li, S.: Improving candidate generation for entity linking. In: Natural Language Processing and Information Systems, pp. 225–236 (2013)
13. Haghighi, A., Klein, D.: Simple coreference resolution with rich syntactic and semantic features. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, vol. 3, pp. 1152–1161. EMNLP (2009)

14. Hajishirzi, H., Zilles, L., Weld, D.S., Zettlemoyer, L.S.: Joint coreference resolution and named-entity linking with multi-pass sieves, pp. 289–299. ACL (2013)
15. Han, X., Zhao, J.: Named entity disambiguation by leveraging Wikipedia semantic knowledge. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 215–224, CIKM 2009 (2009)
16. Hoffart, J., Yosef, M.A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust disambiguation of named entities in text. In: Proceedings of EMNLP, EMNLP 2011, pp. 782–792 (2011)
17. Isele, R., Bizer, C.: Learning expressive linkage rules using genetic programming. PVLDB **5**(11), 1638–1649 (2012)
18. Jiang, L., Wang, J., Luo, P., An, N., Wang, M.: Towards alias detection without string similarity: an active learning based approach. In: SIGIR, pp. 1155–1156 (2012)
19. Kobdani, H.: Linked open government data: lessons from. Institut f*ü*r Maschinelle Sprachverarbeitung (2012)
20. Kulkarni, S., Singh, A., Ramakrishnan, G., Chakrabarti, S.: Collective annotation of Wikipedia entities in web text. In: KDD, pp. 457–466 (2009)
21. Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., Jurafsky, D.: Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, CONLL Shared Task 2011, pp. 28–34 (2011)
22. Lee, H., Recasens, M., Chang, A., Surdeanu, M., Jurafsky, D.: Joint entity and event coreference resolution across documents. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, pp. 489–500 (2012)
23. Lin, T., Mausam, E.O.: No noun phrase left behind: detecting and typing unlinkable entities. In: EMNLP-CoNLL, pp. 893–903 (2012)
24. Mihalcea, R., Csomai, A.: Wikify!: linking documents to encyclopedic knowledge. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, CIKM 2007, pp. 233–242 (2007)
25. Miller, G.A.: Wordnet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
26. Milne, D., Witten, I.H.: Learning to link with Wikipedia. In: Proceedings of Conference on Information and Knowledge Management, CIKM 2009, pp. 509–518 (2008)
27. Technical report. http://www.mpi-inf.mpg.de/yago-naga/aida/
28. Nakashole, N., Tylenda, T., Weikum, G.: Fine-grained semantic typing of emerging entities. In: ACL (2013, to appear)
29. Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., Zhang, Y.: CoNLL-2012 shared task: modeling multilingual unrestricted coreference in ontonotes. In: Joint Conference on EMNLP and CoNLL - Shared Task, pp. 1–40. Association for Computational Linguistics (2012)
30. Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M., Weischedel, R., Xue, N.: CoNLL-2011 shared task: modeling unrestricted coreference in ontonotes. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, CONLL Shared Task 2011, pp. 1–27 (2011)
31. Punyakanok, V., Roth, D., Yih, W., Zimak, D.: Learning and inference over constrained output. In: IJCAI, pp. 1124–1129 (2005). http://cogcomp.cs.illinois.edu/papers/PRYZ05.pdf
32. Parker, R., Graff, D., Kong, J., Chen, K., Maeda, K.: English Gigaword Fifth Edition. Technical reports HPL-2009-155 (2013)

33. Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., Manning, C.: A multi-pass sieve for coreference resolution. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, pp. 492–501 (2010)
34. Rahman, A., Ng, V.: Coreference resolution with world knowledge. In: ACL, pp. 814–824 (2011)
35. Ratinov, L.A., Roth, D.: Learning-based multi-sieve co-reference resolution with knowledge. In: EMNLP-CoNLL, pp. 1234–1244 (2012)
36. Ratinov, L.A., Roth, D., Downey, D., Anderson, M.: Local and global algorithms for disambiguation to wikipedia. In: ACL, pp. 1375–1384 (2011)
37. Singh, S., Subramanya, A., Pereira, F.C.N., McCallum, A.: Large-scale cross-document coreference using distributed inference and hierarchical models. In: ACL, pp. 793–803 (2011)
38. Soon, W.M., Ng, H.T., Lim, D.C.Y.: A machine learning approach to coreference resolution of noun phrases. Comput. Linguist. **27**(4), 521–544 (2001)
39. Spitkovsky, V.I., Chang, A.X.: A cross-lingual dictionary for English Wikipedia concepts. In: LREC, pp. 3168–3175 (2012)
40. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: a core of semantic knowledge. In: WWW, pp. 697–706 (2007)
41. Wick, C.M., Culotta, A., Rohanimanesh, K., Mccallum, A.: An entity based model for coreference resolution (2009)

Mining Credible and Relevant News from Social Networks

Ankur Garg[1](✉), Varun Syal[1], Pankaj Gudlani[1], and Dhaval Patel[2]

[1] Indian Institute of Technology, Roorkee, India
ankurgarg101@gmail.com, varunsyal1994@gmail.com, pankaj.gudlani@gmail.com
[2] IBM TJ Watson Research Center, Yorktown Heights, USA
pateldha@us.ibm.com

Abstract. Today, people are increasingly accessing news through social networks like Twitter. This is regardless of the fact that whether the news is regarding a parliamentary election, or a famous entertainment celebrity. Moreover, these platforms allow people to like, retweet and comment on the shared news article. This shapes the opinions and beliefs of the people who read it along with the news article itself. However, a major problem we face today is the misuse of these networks for spreading rumors and misleading news content. This is the practice of yellow journalism which aims at disrupting public sentiment.

To address this problem, we present a methodology to find credible and relevant tweets that refer to actual news articles published on news websites. Our methodology scores each tweet based on the reputation of the users sharing it, the news publisher which published the news article, and the popularity of the news concepts mentioned in the article. We model the interaction between these three entities in the form of a tripartite graph and propose a Co-HITS algorithm based formulation to score all the entities involved. The scores of individual entities is used to assign a score for each tweet that indicates the credibility and relevance of the news mentioned in it. We find that the presence of many bots is also a big problem in these networks and can affect the results of such explorations. Thus, we use existing bot detection techniques to identify bots and propose an approach to limit their influence on the system in an efficient manner. Finally, we present a qualitative evaluation of our proposed system on a set of approximately 8000 tweets.

Keywords: Data mining · User profiling · Social networks
Twitter · News · Credibility

A. Garg–This author is now at Adobe Research.

V. Syal–This author is now at University of California, San Diego, USA.

P. Gudlani–This author is now at Google Inc.

D. Patel–Work performed when the author was affiliated to Indian Institute of Technology, Roorkee, India.

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 90–102, 2017.
https://doi.org/10.1007/978-3-319-72413-3_6

1 Introduction

News and Social Networks have become two absolutely inseparable phenomenon, deeply interlinked and interdependent. News is the source of information which is disseminated predominantly through social networks. And when it comes to social networks, Facebook and Twitter can be considered as the biggest dispersal platforms for all news in the world. However, this has led to problems like click-bait [1], post-truth politics [2], rumor spreading etc. The ripple effect is highly pronounced in a social network since, it can reach a large number of viewers through dense interconnections almost instantaneously. This plays a role in the formation of users' viewpoints on the different topics of discussion.

In this paper, we aim to find *credible* and *relevant* information from social networks. There have been several attempts, from the research community, in the past to address these problems [3,4]. However, they are mostly supervised, i.e. require annotated ground truth for training of the models to predict *credibility* of information. Such an annotation is very difficult to obtain on a large scale and thus, limits the generality and applicability of the model. Another problem is the presence of bots on Twitter and in general, on any social network. A bot in Twitter is a software that controls the majority of functionalities of a Twitter account using the Twitter API [5,6]. Most of these bots are largely used for spreading false and polarized information. Also, since no human is involved is in the tweeting process, the information spread does not reflect the honest views of a human which is important while reading news from social networks. There are quite a few works in the area of bot detection [5,7–9], but they have not been leveraged to establish the credibility of news and information.

We believe that the social network contains a lot of information within itself, that can be used to address these problems. A news article, based on some *news concepts*, is shared on Twitter, by either the *news source* (which published the article on its website) through its official Twitter account or a user who read the news article and wanted to share his/her comments and opinions along with the original news article. We term such a user as a *newscaster*. Thus, the three entities that play a role in the creation of the tweet are: the *newscasters*, *news sources* and *news concepts*. We posit that the *credibility* of a tweet, mentioning a news article, depends on the popularity of the *news concept* being mentioned as well as the reputation of the *newscaster* and the *news source*. On the other hand, the *relevance* of a tweet can be measured from how recently the news article was published, since as the time of event becomes a thing of past, the news tends to become less relevant. These are the properties that we wish to capture through a tweet scoring function and thus, we formally state our problem statement as: *To propose a scoring function on the set of news tweets such that it can be used to establish a comparison among the tweets on the basis of their credibility and relevance.*

We first develop a methodology to score items of all three entities and then propose a scoring function to quantify the *credibility* and *relevance* of the tweet. The interactions between the three entities can be visualized in the form of a tripartite graph. We employ the Co-HITS algorithm [10] on this graph to obtain a scoring for each of the three sets of entities. Bots present in the system are filtered out using existing bot detection algorithms. Finally, a *tweet score* is given to each tweet, based on the scores given to each of the entities that comprise the tweet. This is the main novelty of our work. Also, an added benefit of the scoring of each entity is that we can get a ranking for each of them. This can help identify the top most *credible newscasters*, *news sources* and *news concepts*. Our contributions in developing this methodology include:

- To develop a scoring function that captures the *credibility* and *relevance* of tweets that refer to news articles (Sect. 4.3).
- Developing a Co-HITS based scoring methodology, on a tripartite graph, to score and rank all the items of the three entities: *newscasters*, *news sources* and *news concepts*(Sect. 4.1).
- To identify bots present in the system using existing techniques and remove them from the system efficiently (Sect. 4.2). User profiling is a time-consuming task and hence, profiling each user to detect whether he is a bot or not is highly inefficient. We thus, propose to profile the *top-k newscasters*, based on the Co-HITS based scoring, and remove them from the system. This ensures that the effect of remaining bots, if any, is negligible.

The rest of the paper is organized as follows. Section 2 describes the prior art while Sect. 3 presents the nature of the dataset used. In Sect. 4, we delineate our solution methodology and present the evaluation results in Sect. 5. Finally, Sect. 6 concludes the paper and mentions the scope of future work in this area.

2 Related Work

Social networks and especially, Twitter have been one of the interesting areas of research in the last decade. The research community has tried to study the different aspects of Twitter like user analysis, content sharing and information credibility.

User analysis refers to analyzing user profiles on Twitter for purposes like influential user discovery [11], recommending users to follow [12] or modeling topical interests of a user [13]. However, a limitation of these works is the fact they use a predefined set of users to model and in our work, the set of crawled headlines gives the set of *newscasters*, for whom an analysis is performed. NCFinder [14] also works off a similar dataset and hence, is a related work for our entity scoring methodology (refer Sect. 4.1). It also models the interactions between the *newscasters*, *news sources* and *news concepts* in the form of a tripartite graph and employs the HITS algorithm [15] to generate scores for all items of the three entities. However, it performs the scoring only based on link information between the entities and does not address the problem of bots present in the system. Further, there have been multiple works that have attempted to identify bots in the Twitter social network based on different attributes of their online activity [5,7–9]. From this, we leverage the work by [5] for filtering undesirable *newscasters* from the system since not only does it consider the tweeting pattern but also analyses the sentiment of the tweet. We believe this to be a crucial indicator of the bots in our system.

Information credibility is another area of related work. [3] first proposed a system that detects whether a tweet related to a trending topic on Twitter is credible or not. This was based on features extracted from the tweet like the citation of external sources, posting behavior of the user and the content of the tweet itself. Another work by [4], performed a credibility analysis in news communities. This was done by using the joint interactions between the users, news and sources to predict the credibility rating given to a news article posted in that community. However, both these works require annotated ground truth which is not explicitly available on Twitter and hence, there is a need to investigate technologies that can work in the absence of such information.

3 Dataset Description

This section delineates the nature of dataset as well as the process in which it is obtained. We also introduce the notations that are used throughout the paper.

We take inspiration from the work of [14] for data collection since we are interested only in tweets related to news published on news websites and working off a fixed user set on Twitter would have limited our study since not users are not expected to tweet about news all the time.

We first crawl a set of headlines, denoted by $\mathcal{H}$, covering a wide range of categories like Politics, Sports, Finance etc. from a list of news websites. The list consisted mostly of popular Indian news websites and some eminent international news websites as well. For each $h \in \mathcal{H}$, we use Stanford Parts-Of-Speech Tagger [16] to tag words present in the headline h. All nouns and collocation of nouns and numerals can be thought of as the potential keywords in the headline, which we formally term as *news concepts*. The news concepts for a headline $h \in \mathcal{H}$ can be denoted by $Cpt(h)$ while the union of all news concepts is referenced by the set $\mathcal{C}$, i.e. $\mathcal{C} = \bigcup_{h \in \mathcal{H}} Cpt(h)$. For example, the headline "Passenger plane crashes in Taiwan with 58 people aboard 9 killed" contains the news concepts: {Passenger plane, Taiwan, 58 people}.

This set of news concepts is used to crawl for the authentic *news tweets* using the Twitter REST API [6]. The set of these authentic tweets is denoted by $\mathcal{T}$ and each tweet in the set contains some concepts in $Cpt(h)$ for some $h \in \mathcal{H}$ as well contains the URL to a real news page containing the same headline h. We use the same URL authentication algorithm mentioned in [14] to verify URLs cited in the tweet and hence, establish authenticity of the tweet itself. Each $t \in \mathcal{T}$ contains the following information as a tuple: *twitter handle* of the user who tweeted this, the *tweet content*, the *timestamp* of the tweet, the *original url* quoted in the tweet, the *headline* in $\mathcal{H}$ about the news pointed by the *original url*, the *retweet count* of the tweet and list of *news concepts* in $\mathcal{C}$, denoted by $tweet\_cpt(t)$, mentioned in the tweet content.

From the set $\mathcal{T}$, we derive two more sets which will be used in the solution description. One is the set of *newscasters* $\mathcal{NC}$ which is the set of all users uniquely identified by the *twitter handle* field present for each $t \in \mathcal{T}$. We collect public attributes for all *newscasters* like: the number of followers and friends. The other set we derive from $\mathcal{T}$ is the set of *news sources* $\mathcal{NS}$ which is the set of all unique domain names identified from the *original url* field for each $t \in \mathcal{T}$.

4 Solution Framework

Figure 1 describes our solution framework. We propose a three step approach to develop a ranking of news tweets based on credibility and relevance. Section 4.1 give the formulation for scoring the entities that comprise a tweet namely, *newscasters*($\mathcal{NC}$), *news sources*($\mathcal{NS}$) and *news concepts*($\mathcal{C}$). Detection and removal of bots is detailed in Sect. 4.2 while the tweet scoring function is proposed in Sect. 4.3.

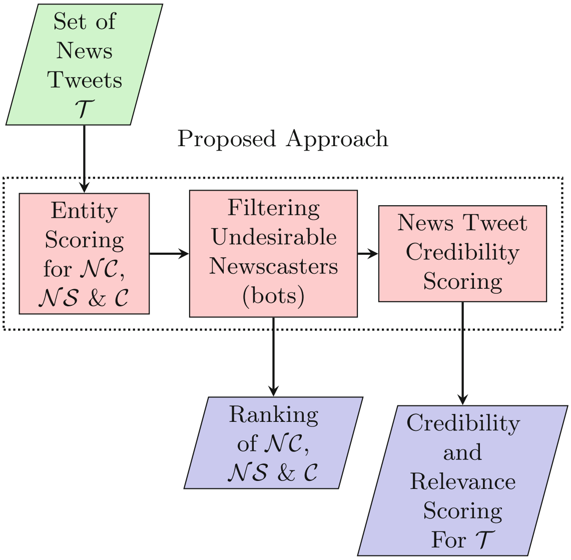

Set of News Tweets $\mathcal{T}$
Proposed Approach
Entity Scoring for $\mathcal{NC}$, $\mathcal{NS}$ & $\mathcal{C}$
Filtering Undesirable Newscasters (bots)
News Tweet Credibility Scoring
Ranking of $\mathcal{NC}$, $\mathcal{NS}$ & $\mathcal{C}$
Credibility and Relevance Scoring For $\mathcal{T}$

Fig. 1. Solution framework

4.1 Entity Scoring

It is interesting to note how each item in any of the three sets, interacts with items of other sets. For example, a *newscaster*, $nc \in \mathcal{NC}$, may have multiple tweets each quoting news from different sources in $\mathcal{NS}$. Similarly, news related to a *news concept*, $cpt \in \mathcal{C}$, may have been published by many *news sources* and tweeted by many *newscasters*. We can visualize these interactions in the form of a weighted tripartite graph, $\mathcal{G}$, where the vertex groups correspond the three sets - $\mathcal{NC}$, $\mathcal{NS}$ *and* $\mathcal{C}$. The weights are defined such that they are proportional to the number of times the two items interact with each other. For $nc \in \mathcal{NC}$ and $ns \in \mathcal{NS}$, $w(nc, ns)$ refers to edge weight between nc and ns in the tripartite graph and can be defined by (1)

$$w(nc, ns) = \frac{|\{t \in \mathcal{T} \mid (nc, ns) \in t\}|}{\sum_{nc \in \mathcal{NC}} \sum_{ns \in \mathcal{NS}} |\{t \in \mathcal{T} \mid (nc, ns) \in t\}|} \quad (1)$$

$w(nc, ns)$ is zero if there no tweet tweeted by nc using the news published by ns. Similarly, the functions $w(nc, cpt)$ and $w(ns, cpt)$ can be defined where $cpt \in \mathcal{C}$.

Moreover, each of the sets have their own inherent properties that can act as an indicator towards its credibility and hence, the news tweet. For a *newscaster*, it can be the number of followers he/she has while for a *news source* it can be metrics computed by reliable external sources like Alexa News Ranking [17]. An ideal scoring method should be able to incorporate this external source of knowledge about the entities as well. It is helpful because we only get a partial view of the whole Twitter network from the data we mine and the knowledge from these external source can lend strength to the findings we make. Consider the case, where we have a *newscaster* who tweets frequently about news but has a low number of followers. A possible reason can be that the news he/she tweets about might be irrelevant and hence, people don't follow the *newscaster*. NCFinder [14] applies the HITS algorithm [15] to the tripartite graph $\mathcal{G}$ described above to score the three sets. We, however propose the use of a variant of HITS, known as Co-HITS [10], to incorporate these inherent properties into the scoring function. The difference from NCFinder is that now, the score given to each item of a set is a weighted sum of its content score (based on an item's inherent properties) and edge weights linking with other entities in the graph $\mathcal{G}$. We run this scoring process for $\mathcal{N}$ iterations when the scores of all items have achieved stability. Equations (2), (3), and (4) describe how the scores are updated. Here $sc(item)_i$ refers to the score of the item in the i-th iteration (i starts from 1).

$$\begin{aligned} sc(nc)_i &= (1 - \lambda_1) * sc(nc)_{i-1} + \lambda_1 * link\_sc(nc)_i \\ link\_sc(nc)_i &= \sum_{ns \in \mathcal{NS}} w(nc, ns) * sc(ns)_{i-1} + \sum_{c \in \mathcal{C}} w(nc, c) * sc(c)_{i-1} \end{aligned} \quad (2)$$

$$\begin{aligned} sc(ns)_i &= (1 - \lambda_2) * sc(ns)_{i-1} + \lambda_2 * link\_sc(ns)_i \\ link\_sc(ns)_i &= \sum_{nc \in \mathcal{NC}} w(nc, ns) * sc(nc)_{i-1} + \sum_{c \in \mathcal{C}} w(ns, c) * sc(c)_{i-1} \end{aligned} \quad (3)$$

$$\begin{aligned} sc(c)_i &= (1 - \lambda_3) * sc(c)_{i-1} + \lambda_3 * link\_sc(c)_i \\ link\_sc(c)_i &= \sum_{ns \in \mathcal{NS}} w(ns, c) * sc(ns)_{i-1} + \sum_{nc \in \mathcal{NC}} w(nc, c) * sc(nc)_{i-1} \end{aligned} \quad (4)$$

$sc(nc)_0$ refers to the initial content score for a *newscaster*. In our case, we set this equal to the relative number of followers a *newscaster* nc has with respect to $\mathcal{NC}$. The Co-HITS [10] algorithm requires that $\sum_{nc \in \mathcal{NC}} sc(nc)_0 = 1$. Similar property should hold for $sc(ns)_0$ and $sc(c)_0$ where $ns \in \mathcal{NS}$ and $c \in \mathcal{C}$. Thus, in any iteration the scores for all items remain between 0 and 1. The weights $\lambda_1,\ \lambda_2,\ \lambda_3$ decide the importance given to the link scores and the content scores. If they are all set to 1.0 then it is equivalent to the standard HITS implementation while a value of 0.0 means the initial scores are the final scores with no link information being considered. Since, we want an equal contribution of the two factors, we set all weights equal to 0.5.

4.2 Filtering Undesirable Newscasters

As explained in Sect. 1, the social networks have been plagued by malicious bots which try to spread biased and false content. When collecting the *news tweets* and building the set $\mathcal{T}$, we perform a URL authentication check (refer Sect. 3) to verify whether the link has news related to the concepts mentioned in the tweet or not. This helps to remove the spam content from our dataset but the bots in the system can still persist. It is because the bots tweet about the same news that they give the link for, but since, these tweets don't reflect any human opinion, they are not credible. Hence, it is important to identify and remove the bots from our system.

To identify bots in our set of *newscasters*, we rely on user profiling system given by [5]. Based on an analysis of the tweets and the profile information of the user, these systems give a *bot score* as output, which is the probability of the account being a bot. Theoretically, we can obtain such scores for all users, but these profiling systems generally take 1–2 mins for one user. On the other hand, our set of *newscasters* is not fixed and may change over time.

Due to this constraint, we only profile the top k *newscasters* as per the scores obtained after the entity scoring in Sect. 4.1. If the *bot score* for any user is higher than 0.7 (chosen after manual inspection of multiple accounts), we label the *newscaster* as a bot and remove the user from the set $\mathcal{NC}$ and his/her tweets from the set $\mathcal{T}$. Furthermore, the sets $\mathcal{NS}$, $\mathcal{C}$ and $\mathcal{G}$ are updated as well to reflect the removal. The value of parameter k can be set either as a fraction of $\mid \mathcal{NC} \mid$ or by using statistical measures like mean and standard deviation to take top *newscasters* above a certain score. The entity scoring is done again to compute the scores on the updated sets of $\mathcal{NC}$, $\mathcal{NS}$ and $\mathcal{C}$.

This step ensures that even if some bots remain in the system, their scores are not high enough which can make a tweet credible and relevant.

4.3 News Tweet Credibility Scoring

Equation (5) gives the tweet scoring function, $tweet\_score(t)$, for each $t \in \mathcal{T}$. Here, $rt(t)$ gives the retweet count of the tweet t while $timestamp(t)$ gives the time when the tweet was published on Twitter. $nc \in \mathcal{NC}$ refers to the *newscaster* and $ns \in \mathcal{NS}$ is the *news source* obtained from the *original url* given in the tweet. $entity\_score(t)$ is a weighted average of the scores of three entities in a tweet, using the weights $w_{\mathcal{NC}}$, $w_{\mathcal{NS}}$ and $w_{\mathcal{C}}$ respectively. Different assignments of the weights has an impact as to which entity influences more in establishing the credibility of the tweet. We examine this impact in the Evaluation section (Sect. 5). *gravity* refers to the exponential decay rate.

$$\begin{aligned} tweet\_score(t) &= \frac{(log(2.0 + rt(t))) * (entity\_score(t))}{(current\_time - timestamp(t))^{gravity}} \\ entity\_score(t) &= \frac{sc(nc) * w_{\mathcal{NC}} + sc(ns) * w_{\mathcal{NS}} + cpt\_score(t) * w_{\mathcal{C}}}{w_{\mathcal{NC}} + w_{\mathcal{NS}} + w_{\mathcal{C}}} \\ cpt\_score(t) &= \frac{\sum_{c \in tweet\_cpt(t)} sc(c)}{\mid tweet\_cpt(t) \mid} \end{aligned} \tag{5}$$

We argue that the scoring function, $tweet\_score(t)$, captures both the relevance and credibility of a tweet $t \in \mathcal{T}$. The denominator is the time decay term inspired from EdgeRank [18] and HackerNews ranking [19] algorithms. This is not a one-time score since the *current_time* changes each time the scoring is done for the same tweet t. Thus, relevance, with respect to time, is ensured since older tweets are bound to get a lower score. This is in line with the fact that older news (and thus, tweets) tend to be less relevant in the present. The decay can be modeled either as linear ($gravity = 1$) or exponential ($gravity > 1$).

Recall, that a tweet t exists in $\mathcal{T}$ as some of its keywords are present in $\mathcal{C}$ (refer Sect. 3). Also, the URL present in the tweet contains a news article which relates to these *news concepts*, and the headline of the article is contained in $\mathcal{H}$. This at least ensures that the information in a tweet is published by some *news source*. The numerator gives a numerical quantification of the credibility of a tweet. If a tweet has been retweeted multiple times, this means that many people agree with the tweet content and the news article. This has been modeled using a logarithmic term to smoothen the effect of large value of retweet counts. $entity\_score(t)$ is the contribution of the scores of entities: *newscasters*, *news sources* and *news concepts*, to the credibility of the tweet. It follows naturally since, if a highly ranked *newscaster* has tweeted the tweet t, then it is highly likely that the news article has been quoted from a reputed *news source*. Also, in this case, the *news concept* being talked about is expected to be a popular one. Similarly, if the *news source*, mentioned in the tweet URL, has a higher score, then the likelihood of the news being credible increases. This can be due to a higher Alexa News Ranking [17] of the *news source* as well due to the fact that better *newscasters* mention it and the published news articles contain popular *news concepts*.

This gives the intuition behind our claim that the scoring function numerically computes the relevance as well as the credibility of a tweet.

5 Evaluation

In this section, we present the results of our proposed approach on the dataset obtained through the process described in Sect. 3. The validation is non-trivial since, to the best of our knowledge no prior art logically equates to our problem.

However, since NCFinder performs entity scoring, we will compare the results of that against our method. For *news tweet* credibility and relevance scoring we perform a qualitative analysis and show that the outputs possess the properties we claim in Sect. 1.

For the evaluation, we selected a subset of the data collected. The subset consists of about 8,000 tweets tweeted on 12th and 13th April 2016. The number of news casters identified in this period were 5,892 tweeting news articles from about 950 news sources. The different news concepts extracted from this tweet corpus were 997.

5.1 Evaluation I: Ranking of Entities Approach

Table 1 shows the top 5 *news sources* and *news concepts*, as per the scores obtained after entity scoring(Sect. 4.1) and filtering of undesirable newscasters (Sect. 4.2). Since, the list of news websites we used to crawl headlines were mostly of Indian origin, hence the top news sources ranked by our approach are from India, which is expected. We, however, omit the names of *newscasters* and identified bots, keeping user privacy in mind.

On comparing the results of our approach with NCFinder, we observe that the use of Co-HITS [10] algorithm improves the quality of ranking of entities obtained. This can be attributed to the use of inherent properties of each entity. For example, a *newscaster* who tweets a large number of news articles published by good *news sources* will get a higher rank through NCFinder. But, if that user has very less number of followers, it can be the case that people don't agree with his view in general and hence, such a user does not appear high in our ranking. We believe this is the strength of our approach.

Another aspect where we perform better than NCFinder is the discovery of bots. The value of k used in the filtering process was taken to be 10% of the size of $\mathcal{NC}$, which comes out to be 600. Out of these, approximately 15% were detected as bots, as the bot scores given by [5] was greater than 0.7. Some of the top 10 *newscasters* found by NCFinder were filtered out as bots which proves the need of the filtering approach. A manual inspection of such accounts showed that, some of the accounts were of real people. But the tweeting pattern and content they shared did not indicate that the user is presenting his views. This can include cases like, just quoting the headline of the news article in the tweet.

Table 1. Results from entity scoring

Top news sources	Top news concepts
www.thehindu.com	India
indiatoday.intoday.in	Health
indianexpress.com	Bernie sanders
timesofindia.indiatimes.com	Law
www.usatoday.com	ipl

We believe it is valid to consider these users as bots and remove them from the system.

5.2 Evaluation II: Relevance Ranking of News Tweets

In (5), we introduce three weights $w_{\mathcal{NC}}$, $w_{\mathcal{NS}}$ and $w_{\mathcal{C}}$. This determines as to how each of the entities- *newscasters*, *news sources* and *news concepts* affect the ranking of tweets.

We used three different weighting schemes to study this impact. In the first scheme, each weight is given a fixed and an equal value, i.e. $w_{\mathcal{NC}} = w_{\mathcal{NS}} = w_{\mathcal{C}} = 1.0$.

In the second weighting scheme, we assigned weights proportionate to the size of the set and its formulation is given by (6).

$$\begin{aligned} w_{\mathcal{NC}} &= \frac{\mid \mathcal{NC} \mid}{\mid \mathcal{NC} \mid + \mid \mathcal{NS} \mid + \mid \mathcal{NC} \mid} \\ w_{\mathcal{NS}} &= \frac{\mid \mathcal{NS} \mid}{\mid \mathcal{NC} \mid + \mid \mathcal{NS} \mid + \mid \mathcal{NC} \mid} \\ w_{\mathcal{C}} &= \frac{\mid \mathcal{C} \mid}{\mid \mathcal{NC} \mid + \mid \mathcal{NS} \mid + \mid \mathcal{NC} \mid} \end{aligned} \tag{6}$$

Finally, we use an inverse-variance weighting scheme given by (7). Here, $\sigma^2_{\mathcal{S}}$ denotes the variance of scores of items in the set S.

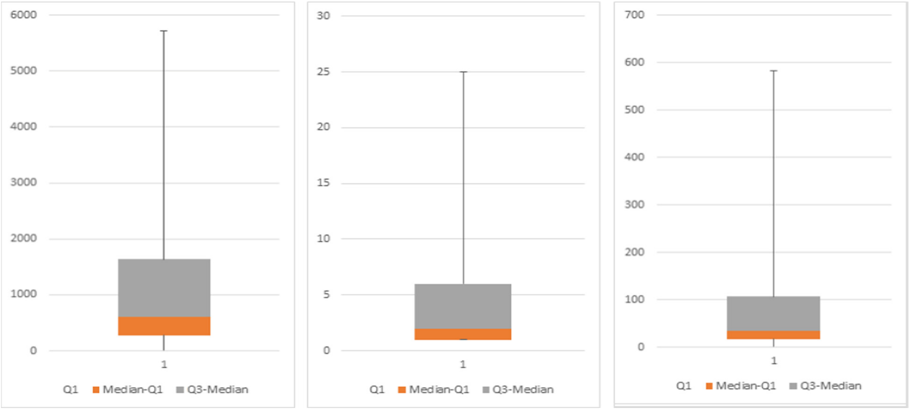
$$\begin{aligned} w_{\mathcal{NC}} &= \frac{1/\sigma^2_{\mathcal{NC}}}{1/\sigma^2_{\mathcal{NC}} + 1/\sigma^2_{\mathcal{NS}} + 1/\sigma^2_{\mathcal{C}}} \\ w_{\mathcal{NS}} &= \frac{1/\sigma^2_{\mathcal{NS}}}{1/\sigma^2_{\mathcal{NC}} + 1/\sigma^2_{\mathcal{NS}} + 1/\sigma^2_{\mathcal{C}}} \\ w_{\mathcal{C}} &= \frac{1/\sigma^2_{\mathcal{C}}}{1/\sigma^2_{\mathcal{NC}} + 1/\sigma^2_{\mathcal{NS}} + 1/\sigma^2_{\mathcal{C}}} \end{aligned} \tag{7}$$

Table 2 shows the values of weights for the different weighting schemes we use.

Table 2. Different weights used for tweet relevance scoring

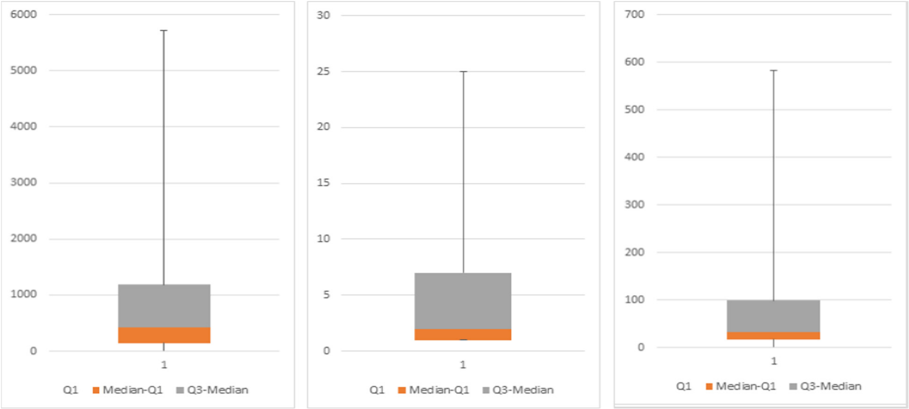
Method of deciding weights	Figure No.	w_{nc}	w_{ns}	w_{cpt}
Equal weights	2	1.0	1.0	1.0
Proportional to set size	3	0.751	0.122	0.127
Inverse variance method	4	0.042	0.041	0.917

Figures 2, 3 and 4 show the box plot distribution of the ranks of the three entities involved in the top 100 most credible and relevant tweets. The different



(a) Newscaster Box plot (b) News Source Box plot (c) News Concepts Box plot

Fig. 2. Rank distribution of the entities using equal weights for all entities

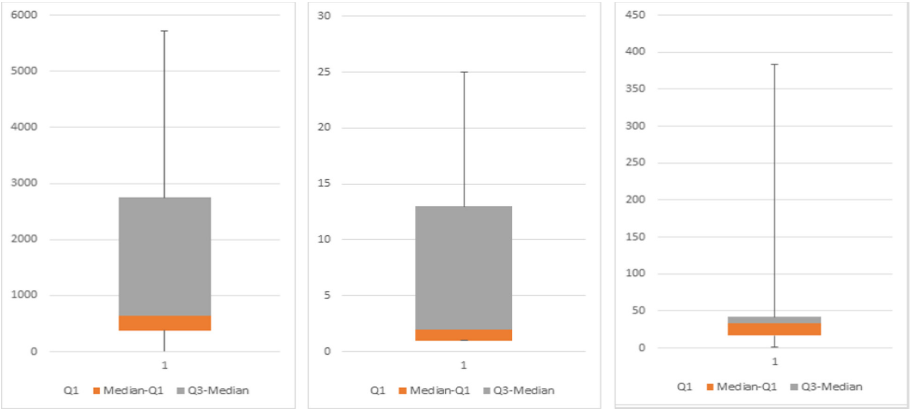


(a) Newscaster Box plot (b) News Source Box plot (c) News Concepts Box plot

Fig. 3. Rank distribution of the entities using weights proportional to entity set sizes

figures correspond to the different weighting schemes employed and summarized in the Table 2.

The difference in distribution is caused by the change in weights. We observe that the distribution of ranks of newscasters and new sources is very similar in all the three schemes. The number of newscasters is very high as compared to the number of news sources and news concepts, and hence, the difference between scores of two newscasters becomes smaller as we go down in the ranking. Thus, newscasters with rank of about 1500 is expected to be very similar to the one at 2000. Hence, in our view inverse variance based weighting scheme seems to perform the best since in comparison to other weighting schemes, the top most news sources and news concepts appear in the most credible and relevant tweets.



(a) Newscaster Box plot (b) News Source Box plot (c) News Concepts Box plot

Fig. 4. Rank distribution of the entities using inverse variance method for deciding the weights

Thus, using this weighted scheme, one can get highly credible and relevant news tweets.

6 Conclusion and Future Work

We introduce a model that scores the news tweets based on their credibility and relevance. Our approach features a novel tripartite Co-HITS algorithm that simultaneously scores three entities: news sources, news casters and news concepts. These scores are then used together to evaluate the relevance and credibility of news tweets. We show that this is a unique problem and our approach provides an efficient scoring function for news tweets. This is a result of taking more effective factors into consideration like using inherent properties of different entities, removing bots and filtering malicious news sources. As an additional outcome, we get a relevance-vise ranking of the current news sources, news casters and news topics.

There is a huge scope to measure information credibility in the absence of labeled data. In the future, we plan to build upon this area to incorporate further sophisticated techniques to completely remove the bots from the system and introduce knowledge bases for fact checking. Another avenue of research can be to develop standard techniques for evaluating works of such kind.

References

1. Visentin L.: Facebook wages war on click-bait. http://www.smh.com.au/digital-life/digital-life-news/facebook-wages-war-on-clickbait-20140825-108dd8.html
2. Viner, K.: How technology disrupted the truth. https://www.theguardian.com/media/2016/jul/12/how-technology-disrupted-the-truth

3. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on twitter. In: Proceedings of the 20th International Conference on World Wide Web, WWW 2011, pp. 675–684. ACM, New York (2011)
4. Mukherjee, S., Weikum, G.: Leveraging joint interactions for credibility analysis in news communities. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM 2015, pp. 353–362. ACM, New York (2015)
5. Ferrara, E., Varol, O., Davis, C.A., Menczer, F., Flammini, A.: The rise of social bots. CoRR, abs/1407.5225 (2014)
6. Twitter: Twitter developer documentation. https://dev.twitter.com/rest/public.
7. Chu, Z., Gianvecchio, S., Wang, H., Jajodia, S.: Detecting automation of twitter accounts: Are you a human, bot, or cyborg? IEEE Trans. Dependable Secure Comput. **9**(6), 811–824 (2012)
8. Dickerson, J.P., Kagan, V., Subrahmanian, V.S.: Using sentiment to detect bots on twitter: Are humans more opinionated than bots? In: 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 620–627. IEEE (2014)
9. Zhang, C.M., Paxson, V.: Detecting and analyzing automated activity on twitter. In: Spring, N., Riley, G.F. (eds.) PAM 2011. LNCS, vol. 6579, pp. 102–111. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19260-9_11
10. Deng, H., Lyu, M.R., King, I.: A generalized co-hits algorithm and its application to bipartite graphs. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009, pp. 239–248. ACM, New York (2009)
11. Cha, M., Haddadi, H., Benevenuto, F., Gummadi, K.P.: Measuring user influence in twitter: The million follower fallacy. In: ICWSM, vol. 10, pp. 10–17 (2010)
12. Hannon, J., Bennett, M., Smyth, B.: Recommending twitter users to follow using content and collaborative filtering approaches. In: Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys 2010, pp. 199–206. ACM, New York (2010)
13. Tao, K., Abel, F., Gao, Q., Houben, G.-J.: TUMS: Twitter-based user modeling service. In: García-Castro, R., Fensel, D., Antoniou, G. (eds.) ESWC 2011. LNCS, vol. 7117, pp. 269–283. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-25953-1_22
14. Mazumder, S., Mehta, S., Patel, D.: Identifying top-k consistent news-casters on twitter. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM 2015, pp. 1875–1878. ACM, New York (2015)
15. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM (JACM) **46**(5), 604–632 (1999)
16. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, NAACL 2003, Vol. 1, pp. 173–180, Stroudsburg. Association for Computational Linguistics (2003)
17. Alexa: Alexa - top sites by category: News. http://www.alexa.com/topsites/category/News.
18. Widman, J.: Edgerank. http://edgerank.net/.
19. Salihefendic, A.: How hacker news ranking algorithm works. https://medium.com/hacking-and-gonzo/how-hacker-news-ranking-algorithm-works-1d9b0cf2c08d.

Integration of Text Classification Model with Speech to Text System

T. S. Aswin(✉), Rahul Ignatius, and Mathangi Ramachandran

Data Science Group, [24]7 iLabs, Bangalore, India
{aswin.ts, rahul.ignatius}@247-inc.com,
ragamudra@gmail.com

Abstract. In the services industry chat helplines were seen as more effective than a voice based service because more number of users could be serviced at the same time and with help of standard text message templates. By training text classifier models and integrating them text to speech conversion systems we can further reduce human effort and thereby deliver efficient solutions with minimal participation and increase user convenience multifold. Our proposed system is the integration of an efficiently trained text classifier model with an open source speech to text conversion plat-form. Our trained model can receive the input in text format from the con-version tool and can accurately classify its category (i.e label it). Based on its classification, consequent action is initiated. Our trained model will eliminate the need for agents manually processing the conversation and initiating required action. The system can save lot of energy, time and other resources.

Keywords: Voice automated text classifier system
Machine learning application integration
User friendly voice assistant model

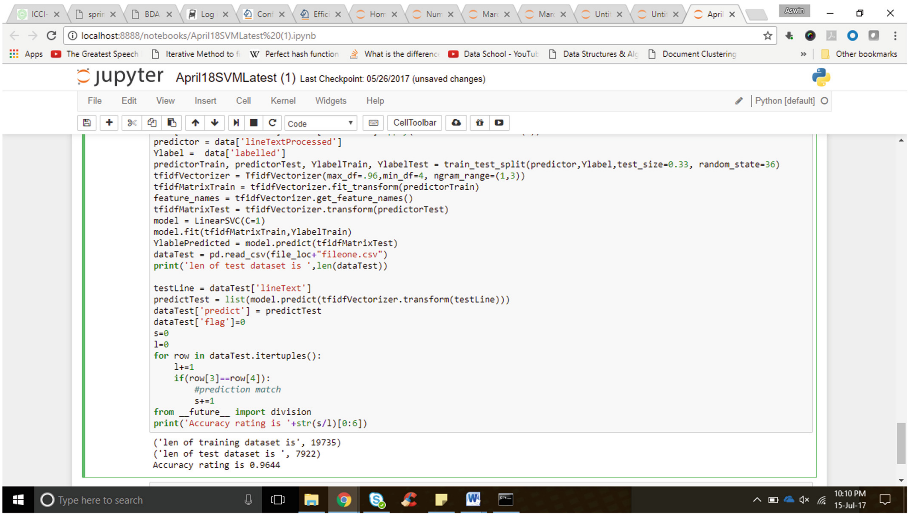
1 Introduction

In the advent of the automation era, it is beneficial to automate all tasks that can be predicted and are of non-complex, non-critical nature [1, 2]. Consider the scenario of customer grievance redressal where most companies establish chat systems or call centers to handle customer grievance messages. When one goes through call transcripts or chat transcripts, it is clear that a wide majority of conversation is initiated for certain purposes that are similar (i.e repetitive in nature) [5, 6]. The benefits of automation are multifold and increasing with advancements in the field [3, 4]. In our project, we shall demonstrate the integration of a text classifier model which has been trained on IVR (Interactive Voice Response) data onto an open source speech conversion platform. In this method we can implement a variety of systems, such as: a primitive talkback assistant, chatbot, voice activated control system and some more services that a talkback assistant can support.

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 103–112, 2017.
https://doi.org/10.1007/978-3-319-72413-3_7

2 Background Information

Large dataset of IVR of customer grievances were manually labeled over a week, for creating a labeled dataset, since a supervised approach is followed. One dataset for training and another dataset were also tagged for testing purposes. A text classifier model has been trained using Support Vector Machines (SVM) primarily because of the high efficiency with regards to labeling. The accuracy levels were noted to be around 96% (see Fig. 1 below). Since our model yields a high accuracy score, we can reliably use it for classification of IVR lines submitted to it. What happens is that we can build a model, pickle the file (i.e store it for future use). The pickled file contains the relevant features to be observed and the function to evaluate new lines.



localhost:8888/notebooks/April18SVMLatest%20(1).ipynb
jupyter April18SVMLatest (1) Last Checkpoint: 05/26/2017 (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Python [default]
Code
CellToolbar
Type here to search
10:10 PM
15-Jul-17

```
predictor = data['lineTextProcessed']
Ylabel =  data['labelled']
predictorTrain, predictorTest, YlabelTrain, YlabelTest = train_test_split(predictor,Ylabel,test_size=0.33, random_state=36)
tfidfVectorizer = TfidfVectorizer(max_df=.96,min_df=4, ngram_range=(1,3))
tfidfMatrixTrain = tfidfVectorizer.fit_transform(predictorTrain)
feature_names = tfidfVectorizer.get_feature_names()
tfidfMatrixTest = tfidfVectorizer.transform(predictorTest)
model = LinearSVC(C=1)
model.fit(tfidfMatrixTrain,YlabelTrain)
YlablePredicted = model.predict(tfidfMatrixTest)
dataTest = pd.read_csv(file_loc+"fileone.csv")
print('len of test dataset is ',len(dataTest))

testLine = dataTest['lineText']
predictTest = list(model.predict(tfidfVectorizer.transform(testLine)))
dataTest['predict'] = predictTest
dataTest['flag']=0
s=0
l=0
for row in dataTest.itertuples():
    l+=1
    if(row[3]==row[4]):
        #prediction match
        s+=1
from __future__ import division
print('Accuracy rating is '+str(s/l)[0:6])
```

```
('len of training dataset is', 19735)
('len of test dataset is ', 7922)
Accuracy rating is 0.9644
```

Fig. 1. Code snippet showing the accuracy of the trained text classifier

There is a need for a highly trained model so that the white noise (i.e. unnecessary features in the data) is disregarded and only the required features are picked up by the model. In case the model is trained on unnecessary features, misclassification may happen. Integrating the speech to text system with the text classifier can be construed as a convenience factor, so that the user does not have to type in the text for the model to process it. An efficient speech to text system will ensure the user's effort is minimized and the service is more user-centric and personalized (Table 1).

2.1 Choice of Model

Since we wished to build a model with optimum efficiency, we ruled out the unsupervised machine learning methods followed and took up Support Vector Machines to

Table 1. Sample lines with their labels

IVR line	Classification
I need to reset my password	Login-get-help
I forgot my password	Login-get-help
Trouble logging in	Login-get-help
I think my account has been hacked	Unauthorized-use-report
Unauthorized payment	Unauthorized-use-report
My account has been compromised	Unauthorized-use-report
My account has been blocked	Account-suspended-get-help
Why has my account been suspended	Account-suspended-get-help
My account is suspended	Account-suspended-get-help

build a scalable, efficient model. The dataset was refined of stop words (trivial words like “me”, “my”, “i”, etc. that aren’t needed for the model) and other unwanted markers like profanity, location, number and location [11, 12]. A large dataset was used as training data, and another for testing the model’s efficiency. Following the Bag of Words model, the records were split into new records of 2–3 words each (following the N-gram approach).

Finding the efficiency rating of the trained classifier model to be satisfactory, it was decided to integrate the model with a Text To Speech (TTS) system so as to build a system that can offer more user convenience and efficiency than currently available customer facing systems [7, 8]. Our system is hosted on a cloud machine. By leveraging the use of open source tools, we have made use of an efficient open source speech to text (TTS) system that can complement our system [9, 10].

2.2 System Design

Our system is hosted as a server program on a remote computer. If any user wishes to obtain classification for some IVR lines (i.e make a query), a request should be made to the computer hosting the server program (i.e. establish a connection) and then invoke the required function by passing the IVR line as a parameter. Figure 2 here shows workflow of earlier systems.

Our system’s design is seen below in Fig. 3. Almost all devices we use today support the function of audio collection, and playing. So we can extend our program to a larger user base at no extra costs to the user, while ensuring better convenience aspects, than having the user call, wait in queue, or login to a secure account and type the query to chat with the agent. The resources of the host are also saved, in terms of time, investments in human resource, platforms, connectivity charges and other related

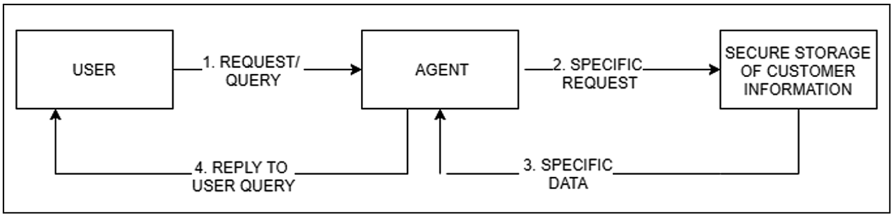


Fig. 2. A stepwise block diagram of the earlier existing system

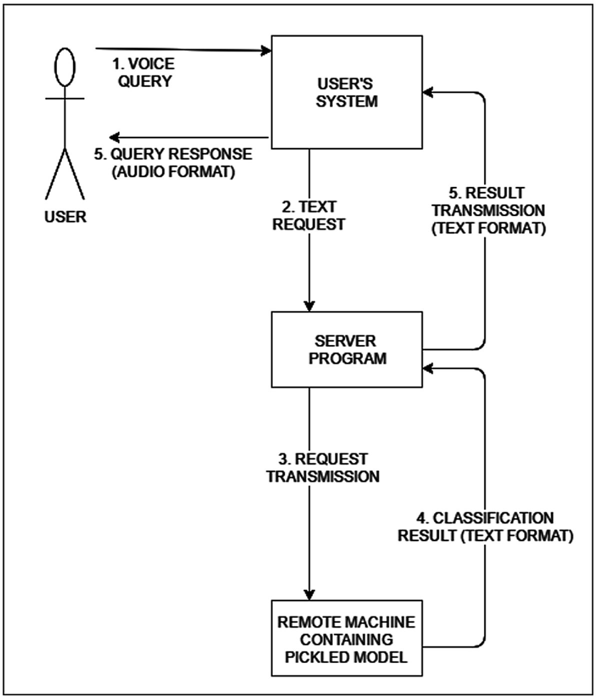
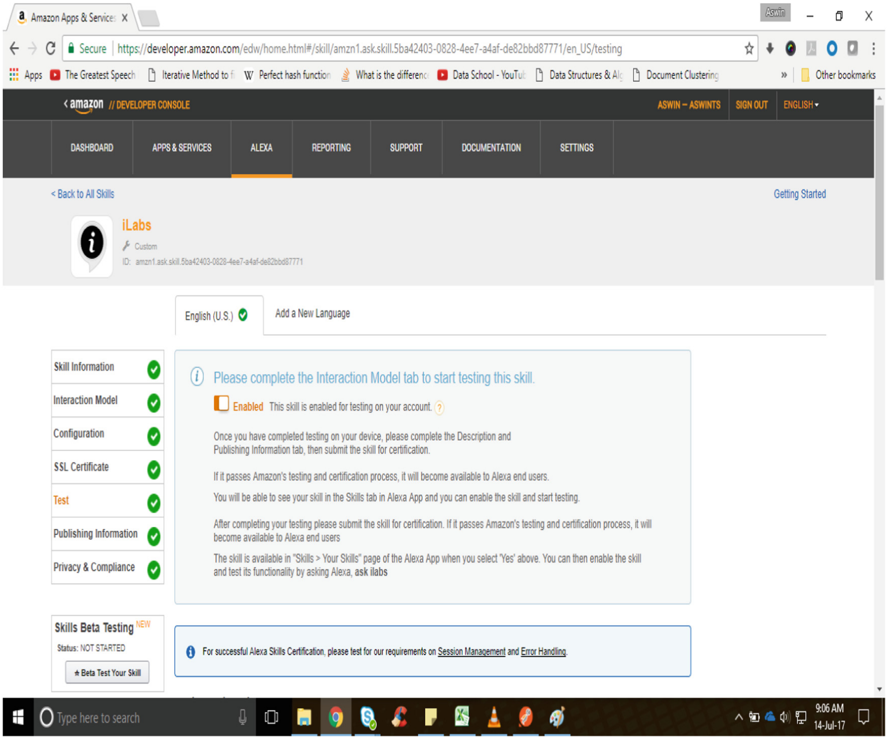


Fig. 3. Our model workflow sequence with a Text To Speech (TTS) system

costs. The future of service industries is said to largely belong to advancements in automation, while ensuring the systems are reliable, safe and multiple times more efficient than having it done manually.

3 System Implementation

Observing our model workflow in Fig. 3, let us call the remote machine as the ‘target machine’ and the server as the ‘man in the middle’ machine. The ‘man in the middle’ machine is required to handle traffic, balance loads, and direct requests to the required target machine (in case any one target machine is handling large volume of traffic). In case a single target machine can only handle a certain number of queries at a particular unit of time, the ‘man in the middle’ machine, or routing machine, can direct the queries to an alternate target machine. For implementation purposes we have used the Amazon Alexa Speech toolkit and hosted a skill on their platform (see Fig. 4 below). This platform is one of the many available open source platforms where users can host their applications. Amazon Alexa is primarily a speech to text system that directs our oral queries or commands to the corresponding action centre. Here we leverage this speech to text system to direct oral queries to a program that we have hosted on a cloud machine.



Please complete the Interaction Model tab to start testing this skill.

Enabled This skill is enabled for testing on your account.

Once you have completed testing on your device, please complete the Description and Publishing Information tab, then submit the skill for certification.

If it passes Amazon's testing and certification process, it will become available to Alexa end users.

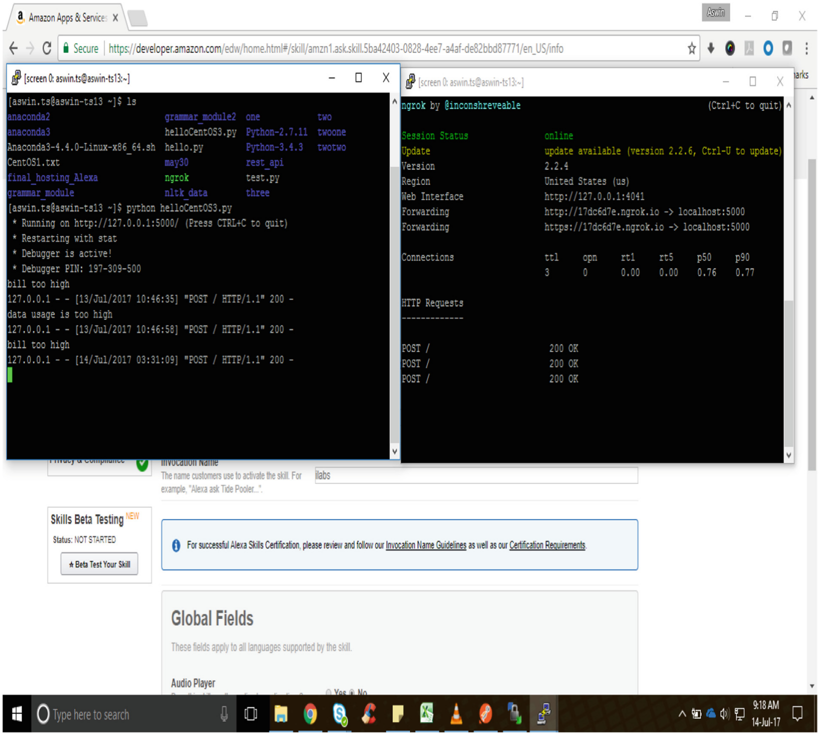
You will be able to see your skill in the Skills tab in Alexa App and you can enable the skill and start testing.

After completing your testing please submit the skill for certification. If it passes Amazon's testing and certification process, it will become available to Alexa end users

The skill is available in "Skills > Your Skills" page of the Alexa App when you select 'Yes' above. You can then enable the skill and test its functionality by asking Alexa, **ask ilabs**

For successful Alexa Skills Certification, please test for our requirements on Session Management and Error Handling.

Fig. 4. Developing a speech program on Amazon Alexa developer console



```
[aswin.ts@aswin-ts13 ~]$ ls
anaconda2                        grammar_module2  one            two
anaconda3                        helloCentOS3.py  Python-2.7.11  twoone
Anaconda3-4.4.0-Linux-x86_64.sh  hello.py         Python-3.4.3   twotwo
CentOS1.txt                      may30            rest_api
final_hosting_Alexa              ngrok            test.py
grammar_module                   nltk_data        three
[aswin.ts@aswin-ts13 ~]$ python helloCentOS3.py
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 197-309-500
bill too high
127.0.0.1 - - [13/Jul/2017 10:46:35] "POST / HTTP/1.1" 200 -
data usage is too high
127.0.0.1 - - [13/Jul/2017 10:46:58] "POST / HTTP/1.1" 200 -
bill too high
127.0.0.1 - - [14/Jul/2017 03:31:09] "POST / HTTP/1.1" 200 -
```

```
ngrok by @inconshreveable                                   (Ctrl+C to quit)

Session Status                online
Update                        update available (version 2.2.6, Ctrl-U to update)
Version                       2.2.4
Region                        United States (us)
Web Interface                 http://127.0.0.1:4041
Forwarding                    http://17dc6d7e.ngrok.io -> localhost:5000
Forwarding                    https://17dc6d7e.ngrok.io -> localhost:5000

Connections                   ttl     opn     rt1     rt5     p50     p90
                              3       0       0.00    0.00    0.76    0.77


HTTP Requests
-------------


POST /                         200 OK
POST /                         200 OK
POST /                         200 OK
```

The name customers use to activate the skill. For example, "Alexa ask Tide Pooler...".

For successful Alexa Skills Certification, please review and follow our Invocation Name Guidelines as well as our Certification Requirements.

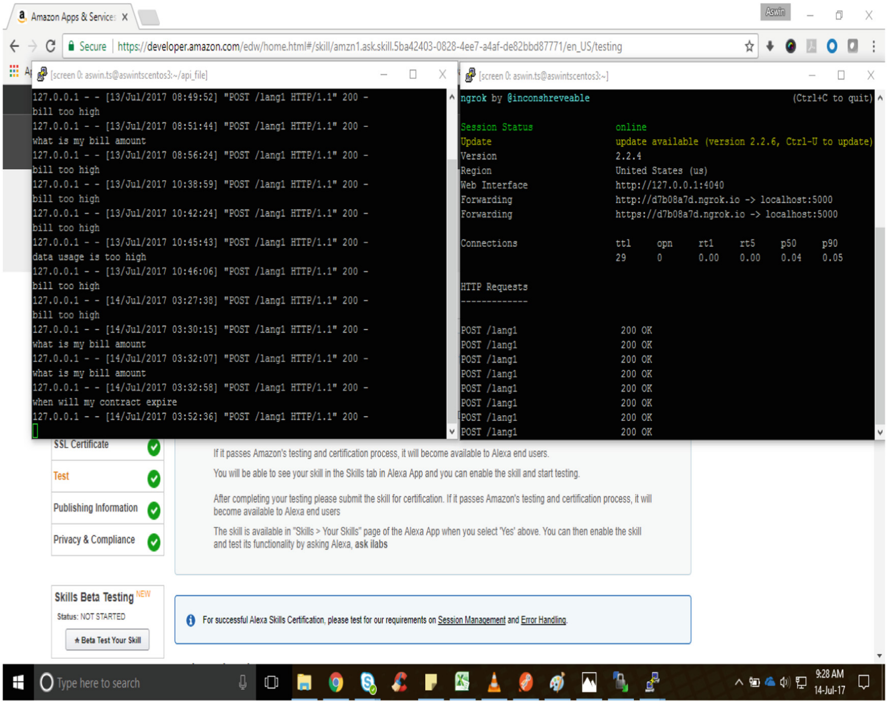
These fields apply to all languages supported by the skill.

Fig. 5. Hosting the 'man in the middle' system on a cloud machine

An Open Source proxy software called 'NGROK' was used to create a secure tunnel to our cloud machine from the public endpoint. Then we need to host the 'man in the middle' machine and the target machine. Another open source tool has been used, called 'Postman' to periodically test that the 'target' machine is active and responding efficiently and accurately to queries.

The best use of Open source tools have been made to leverage large number of users on the existing speech platform, the Alexa platform, from refining the data to hosting the remote machines. By leveraging approved open source tools we are using these tools to complement our system and therefore function with highest standards.

So the 'man in the middle' machine is part of the scalability aspect of the system to ensure that all queries are given roughly the same importance and there may not be a single point of failure. The design may be done in such a way that if responses are not received within a certain time frame, the unanswered requests may be directed to available target machines. (see Fig. 5 below) The 'man in the middle' machine may also be used to keep track of the requests for multiple purposes such as log keeping,


Amazon Apps & Services
Aswin
Secure
https://developer.amazon.com/edw/home.html#/skill/amzn1.ask.skill.5ba42403-0828-4ee7-a4af-de82bbd87771/en_US/testing
[screen 0: aswin.ts@aswintscentos3:~/api_file]
[screen 0: aswin.ts@aswintscentos3:~]
SSL Certificate
Test
Publishing Information
Privacy & Compliance
Skills Beta Testing NEW
Status: NOT STARTED
★ Beta Test Your Skill
Type here to search
9:28 AM
14-Jul-17

```
127.0.0.1 - - [13/Jul/2017 08:49:52] "POST /lang1 HTTP/1.1" 200 -
bill too high
127.0.0.1 - - [13/Jul/2017 08:51:44] "POST /lang1 HTTP/1.1" 200 -
what is my bill amount
127.0.0.1 - - [13/Jul/2017 08:56:24] "POST /lang1 HTTP/1.1" 200 -
bill too high
127.0.0.1 - - [13/Jul/2017 10:38:59] "POST /lang1 HTTP/1.1" 200 -
bill too high
127.0.0.1 - - [13/Jul/2017 10:42:24] "POST /lang1 HTTP/1.1" 200 -
bill too high
127.0.0.1 - - [13/Jul/2017 10:45:43] "POST /lang1 HTTP/1.1" 200 -
data usage is too high
127.0.0.1 - - [13/Jul/2017 10:46:06] "POST /lang1 HTTP/1.1" 200 -
bill too high
127.0.0.1 - - [14/Jul/2017 03:27:38] "POST /lang1 HTTP/1.1" 200 -
bill too high
127.0.0.1 - - [14/Jul/2017 03:30:15] "POST /lang1 HTTP/1.1" 200 -
what is my bill amount
127.0.0.1 - - [14/Jul/2017 03:32:07] "POST /lang1 HTTP/1.1" 200 -
what is my bill amount
127.0.0.1 - - [14/Jul/2017 03:32:58] "POST /lang1 HTTP/1.1" 200 -
when will my contract expire
127.0.0.1 - - [14/Jul/2017 03:52:36] "POST /lang1 HTTP/1.1" 200 -
```

```
ngrok by @inconshreveable                                   (Ctrl+C to quit)

Session Status                online
Update                        update available (version 2.2.6, Ctrl-U to update)
Version                       2.2.4
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    http://d7b08a7d.ngrok.io -> localhost:5000
Forwarding                    https://d7b08a7d.ngrok.io -> localhost:5000

Connections                   ttl     opn     rt1     rt5     p50     p90
                              29      0       0.00    0.00    0.04    0.05


HTTP Requests
-------------


POST /lang1                    200 OK
POST /lang1                    200 OK
POST /lang1                    200 OK
POST /lang1                    200 OK
POST /lang1                    200 OK
POST /lang1                    200 OK
POST /lang1                    200 OK
POST /lang1                    200 OK
```

If it passes Amazon's testing and certification process, it will become available to Alexa end users.

You will be able to see your skill in the Skills tab in Alexa App and you can enable the skill and start testing.

After completing your testing please submit the skill for certification. If it passes Amazon's testing and certification process, it will become available to Alexa end users

The skill is available in "Skills > Your Skills" page of the Alexa App when you select 'Yes' above. You can then enable the skill and test its functionality by asking Alexa, **ask ilabs**

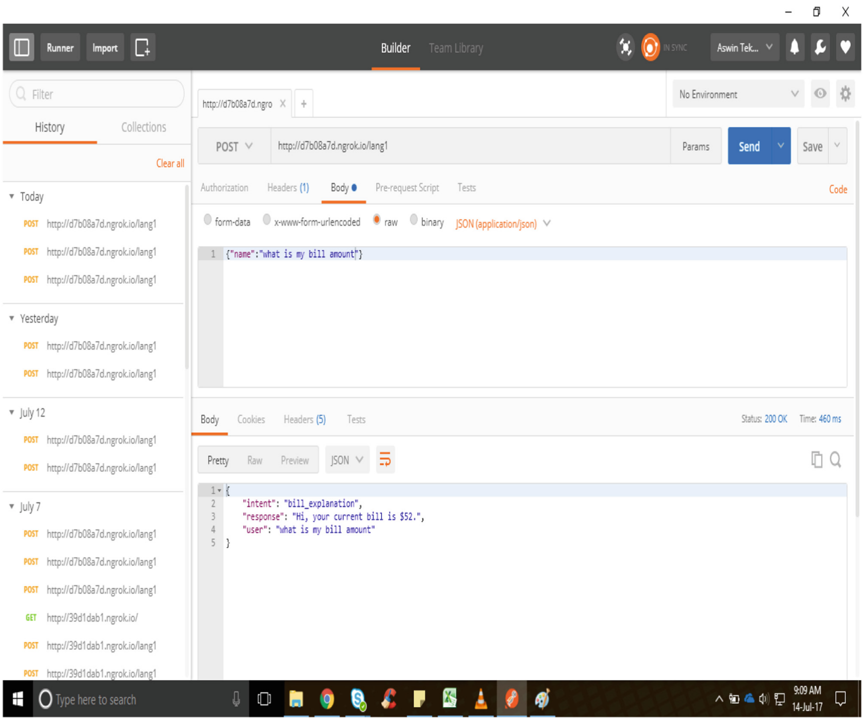
For successful Alexa Skills Certification, please test for our requirements on Session Management and Error Handling.

Fig. 6. The target machine has been tested to run optimally on a remote machine

billing, load balancing, network connectivity monitoring, power management(if rate of requests incoming are very low, the multiple target machines may be powered down and the 'man in the middle' machine can itself act as the target machine) and so on.

The testing in our case was done in phases, i.e. the model accuracy was tested first, then the model was tested on a local machine, then tested with integration with the speech to text system, then sample programs were hosted on cloud systems to check that they are responsive and agile. As mentioned before, open source software was used to host programs on cloud machines (see Fig. 6 below). Once the cloud systems responded efficiently, the program was hosted on a remote machine.

Furthermore, open source tools were used during troubleshooting phase, to ensure that any errors caused were not due to network or other connectivity issues. In such scenarios, freely available tool named 'Postman' was used to check periodically that the program is responding to sample queries(see Fig. 7 below). Once the 'man in the middle' machine was checked and found to be responsive, tests were repeated for the target machine too.



Today
POST http://d7b08a7d.ngrok.io/lang1
POST http://d7b08a7d.ngrok.io/lang1
POST http://d7b08a7d.ngrok.io/lang1
Yesterday
POST http://d7b08a7d.ngrok.io/lang1
POST http://d7b08a7d.ngrok.io/lang1
July 12
POST http://d7b08a7d.ngrok.io/lang1
POST http://d7b08a7d.ngrok.io/lang1
July 7
POST http://d7b08a7d.ngrok.io/lang1
POST http://d7b08a7d.ngrok.io/lang1
POST http://d7b08a7d.ngrok.io/lang1
GET http://39d1dab1.ngrok.io/
POST http://39d1dab1.ngrok.io/lang1
POST http://39d1dab1.ngrok.io/lang1

POST http://d7b08a7d.ngrok.io/lang1

Body: raw JSON (application/json)

```
{"name":"what is my bill amount"}
```

Body Status: 200 OK Time: 460 ms JSON

```
{
    "intent": "bill_explanation",
    "response": "Hi, your current bill is $52.",
    "user": "what is my bill amount"
}
```

Fig. 7. Testing the target machine with an open source testing tool called 'Postman'

4 Conclusion

Thus we have explored the option of alternating to a system where user can very conveniently place queries to a well-designed speech to text platform that will direct the query to our system for further processing. The classifier model then efficiently returns the reply to the speech platform which returns the content in an audio format.

It has been interesting to note that our only major investments were of time: in labeling the data and network: of hosting the classifier model and optionally another cloud machine for handling and scheduling queries (i.e a focus on being a large scalable service system). We leveraged available open source tools and focused only on improving our model accuracy and succeeded in building a low cost model by leveraging third party Open Source software to accomplish tasks such as hosting, testing connectivity and extending system functionality to speech to text. For services of operational nature, it is usually recommended to retrain the model periodically with different data, optimally the most recent possible.

5 Future Applications

1. Consider the scenario of a person at home who wishes to perform certain trivial tasks, like drawing the curtains, switching on or off the lights, operating the TV system, the stereo and similar services. In such cases it is essential that the control system captures the user's commands as accurately as possible.
2. Operating robots, handling electronic appliances, security systems, voice activated controls on automobiles, etc. The applications of user convenient voice activated problems are included.
3. We have just built a primitive talkback assistant since we can issue commands based on our voice messages. When adequately trained, the system can reduce our manual efforts by a large margin. Support for third party applications could ensure we can even access applications, content stored in our devices or online and perform or activate certain actions.
4. We can even build primitive search engines where the user can access certain data or have it sent to his/ her system. A step further, it could also be a filter based system for kids, or in a restricted environment where the searches could be filtered and certain key-phrases could trigger alarms. Any similar functionality can be supported.

6 Challenges in Implementation

1. It is essential that the model which is implemented (built by supervised or unsupervised machine learning) is optimally conditioned for best results. In our approach we have assumed no manual error in labeling the data in our training dataset.
2. For scalability, it is essential that the deployed system is optimal and capable of handling large number of requests simultaneously.
3. It is recommended to use free and open source services, in our instance for servers, and Text to Speech systems that are available than paid services, since one of the primary aims of building automated systems is to cut costs but still maintain a high rate of efficiency for the program. Moreover, with Open Source systems the documentation is usually well maintained and supported by a large user base.
4. A number of tertiary factors arise in Text to Speech systems, such as the language used, the context etc. Certain limitations in the advancement of Natural Language Processing (NLP) technologies exist in some languages, dialects, geographic region wise, and the capability of the system to recognize the user's spoken words. It is expected that with advances in NLP in the future, we can extract more information from language content.
5. Small error margins exist in automation and classification, since accuracy results noted are not fully capable of correctly identifying input lines, which is one of the few reasons why such systems are not implemented in critical areas like hospital patient management, critical divisions of law enforcement operations, top security establishments, and defense systems. In such cases, automated classification may be

used, but only to complement the physical presence of personnel. With advances in the field of automation in the near future, it is expected that nearly cent percent reliable systems can be designed and deployed for handling critical tasks.

Acknowledgement. The authors would like to thank the Research and Development division of [24]7 Bangalore for the data and resources that made this project possible.

References

1. Business Standard: Automation to replace work for repetitive work, says report, 16 September 2016. http://www.business-standard.com/article/companies/automation-to-replace-people-for-repetitive-work-says-report-116091400729_1.html
2. Business Standard: 89% people positive on benefits of automation, robots in workplace: study, 17 February 2017. http://www.business-standard.com/article/current-affairs/89-people-positive-on-benefits-of-automation-robots-in-workplace-study-117021700324_1.html
3. The Economic Times: Artificial Intelligence could turn some skilled practices into utilities: Gartner, 9 May 2017. http://cio.economictimes.indiatimes.com/news/business-analytics/artificial-intelligence-could-turn-some-skilled-practices-into-utilities-gartner/58592427
4. BBC: How automation could benefit agriculture, 16 September 2015. http://www.bbc.com/news/science-environment-34271384
5. Forbes: Impact of automation on the independent workforce, 2 May 2017. https://www.forbes.com/sites/forbeshumanresourcescouncil/2017/05/02/the-impact-of-automation-on-the-independent-workforce/#bc1366475c51
6. Forbes: Automation and the future of work, 2 December 2016. https://www.forbes.com/sites/adigaskell/2016/12/22/automation-and-the-future-of-work/#68e1198471fc
7. Bijl, D., Hyde-Thompson, H.: Speech to text conversion. U.S Patent No. 6,173,259, 9 January 2001
8. Riccardi, G., Hakkani-Tur, D.: Active learning: theory and applications to automatic speech recognition. IEEE Trans. Speech Audio Process. **13**(4), 504–511 (2005)
9. Taylor, P.: Text to Speech Synthesis. Cambridge University Press, Cambridge (2009)
10. Vermeulen, P., Mozer, T.F.: Client/server architecture for text to speech synthesis. U.S. Patent No. 6,810,379. 26 October 2004
11. Kao, A., Poteet, S.R. (eds.): Natural Language Processing and Text Mining. Springer, London (2007). https://doi.org/10.1007/978-1-84628-754-1
12. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0026683

Modern Column Stores for Big Data Processing

K. T. Sridhar[1,2(✉)]

[1] XtremeData Technologies, Bangalore, India
sridhar@xtremedata.com
[2] XtremeData, Inc., Schaumburg, USA

Abstract. The advent of MapReduce/Hadoop and NoSQL databases undermined the primacy of SQL relational databases for data processing. Pioneering work by researchers on MonetDB and C-Store opened up the world of column stores that retain the SQL model but use different store and engine for performance gains. The emergence of pay-by-use clouds and MPP versions of column stores on cloud eliminated scale-out issues of row stores. Data mining researchers have also shown that SQL on parallel, columnar database could be a candidate for Big Data analytics. In this survey written for a tutorial, we trace the technology evolution and history of the *fall* of row stores and *rise* of column stores, delving into architectural details of column DBs from academia and industry.

Keywords: Column store · SQL · NoSQL · Big Data · Data mining

1 Introduction

Since the days of System R [1] and its product version SQL/DS of IBM, relational databases have dominated data processing. The inventor of relational model Edgar Codd in his Turing Award lecture terms databases as "*a practical foundation for productivity*". System R and its descendants used a row store organized as fixed size pages on disk, and executed SQL queries using an *iterator* model, which was a tuple-at-a-time interpreter with a simple abstract interface of *open*, *next* and *close* calls [2].

Initially, SQL DBMSs targeted enterprise business applications and were successful in becoming the *numero uno* tool for OLTP applications in banking, enterprise information systems including ERP, SCM, CRM, etc. Later, to deal with ad-hoc queries on larger data volumes in business intelligence and OLAP applications, they metamorphosed into *massively parallel processing (MPP)* systems, running on a cluster of nodes distributing data amongst the nodes, and using either *shared disk* or *shared nothing* storage.

Newer SQL features for multidimensional modeling through data cubes, materialized views, statistical aggregates, moving window aggregates, etc., and ETL tools for data quality [3] were added. Industry innovated further: custom FPGA hardware attached to disks for query processing on a row store, MPP, *database appliance* for OLAP, and even OLTP applications, from Netezza.

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 113–125, 2017.
https://doi.org/10.1007/978-3-319-72413-3_8

Despite several advances in row store DBMSs, today they have lost their vice-like grip on data processing, and have become legacy systems for OLTP applications that are now being threatened by a newer wave of ACID compliant NewSQL [4] products. In the rest of this paper we examine the reasons for this fall from a pre-eminent position and the disruptive technology advances that brought in column stores (Sect. 2), delve into internals of column store databases (Sect. 3) from academia and industry, illustrate the use of SQL on parallel, column stores for structured Big Data processing from published literature (Sect. 4), and conclude, somewhat optimistically, in Sect. 5.

2 Background

We trace the technology evolution and history that triggered the fall of row store databases and rise of column stores, and place it in perspective with respect to advances in related domains that catalyzed the churn in data processing.

2.1 Dawn of Big Data Era

The meteoric rise of the internet loosened the vice-like grip of relational databases on data processing for web-scale data and it's mining. On democratization and scaling-up of the web along with the onset of Web 2.0 and mobile devices that are now ubiquitous, the Big Data [5] era was upon us. The grip weakened further, and Big Data ushered in newer technologies: NoSQL [6] to NewSQL [4].

Google invented MapReduce [7], a programming model for distributed applications inspired by LISP like functional programming but with imperative languages. Apache developed open source Hadoop, a distributed file system that provided the infrastructure for MapReduce, simplifying distributed application development by relieving the programmer from managing such applications including the handling of failures.

A number of applications adopted the MapReduce/Hadoop paradigm for data processing. There was a proliferation of NoSQL products based on key/value store (Dynamo, Voldemort), wide-column stores (Bigtable, HBase, Cassandra), documents (MongoDB, CouchDB), graphs (Neo4j, Giraph), etc. Details of these products may be found in the comprehensive survey [6] and also [4,5].

2.2 Friends Turn Foes!

Almost contemporaneous to the arrival of MapReduce, database researchers too were investigating issues with relational row stores at several levels: hardware, transactions, applications, etc.

Even before large scale web mining and Big Data, Ailamaki et al. [8] investigated the inability of DBMS products, unlike HPC systems, to take advantage of major advances in hardware technology. They evaluated four row store DBMS products against a synthetic benchmark and TPC-H to conclude that 50% of query processing time was taken up by CPU stalls. Query processing algorithms were not CPU cache aware, neither for data nor for instructions. The nice and simple iterator model with tuple-at-a-time processing needed a revamp.

Brewer proposed the CAP theorem [9]: a network shared-data system can support only two of three desirable properties, *consistency* (C), *high availability* (A) and *partition tolerance* (P); essentially a negative result for trade-offs in distributed systems. CAP theorem justified the compromises made by NoSQL systems with reference to consistency, leading to a different view of transactions: *BASE* for *basically available* (BA), *soft-state* (S) and *eventual consistency* (E).

In the context of data processing applications, Stonebraker, a lead architect of Ingres from System R days and 2014 Turing Award winner, raised the question of whether "*one size fits all*"? The authors of [10] based their arguments on examples of data processing applications that varied widely in characteristics and query processing requirements:

- *stream processing*: sensor network, financial-feed, algorithmic trading, etc. with low latency processing
- *data warehousing*: OLAP not suited for write optimized systems
- *scientific databases*: astronomy, particle physics, etc. require array processing
- *text data*: library information systems, medical records, legal systems, web pages, etc. with semi-structured and/or unstructured data

They categorize DBMSs as *outbound* systems that must write before processing; low latency applications need *inbound* systems like stream databases, very different from row store DBMS. They advocate use of domain specific DB engines and turn soothsayers: "*'one size fits all' theme is unlikely to successfully continue under these circumstances*". Indeed, their prediction has come true today and the all-encompassing, monolithic approach of row store DBMSs is no longer the norm: the gecko has fallen!

2.3 Lost in the Woods

Traditional row store DBMSs became *persona non grata* for processing Big Data, defined by 3Vs (*volume*, *velocity*, *variety*) [5] and later enhanced with more Vs (*veracity*, *variability*, *value*). The general perception was that SQL databases were inadequate [11] to deal with the original 3Vs.

- dealing with volume required easy horizontal scalability as in NoSQL systems; the recommended solution of DBMS vendors of vertical scale-up to deal with volume was too expensive.
- as highlighted by [10] handling high velocity data required low latency processing, which DBMSs did not do.
- databases were designed to deal with structured data like numbers, strings, etc., not semi-structured data common in the web (XML, JSON) or unstructured data like audio, video, tweets, etc.

What of structured data that DBMSs were good at processing? Even in the realm of structured Big Data, DBMSs became marginal for a variety of reasons:

(1) not able to deal with structured Big Data volumes (2) performance issues with row stores (3) too many knobs (4) cost factors: database appliances were too expensive and most NoSQL systems were open source and (5) social factors: everyone wanted to take the high road to be on the crest of the NoSQL wave!

2.4 Light at the End of the Tunnel?

Motivated by advances in hardware technology, and inspired by work of Ailamaki et al. [8], a group of researchers at CWI, The Netherlands, abandoned the row store and built an ACID compliant column store database, *MonetDB* [12] that partitioned data vertically across columns of a table. They also devised a query processing layer that did not use the iterator model and took advantage of modern hardware with innovative algorithms. MonetDB was open sourced and research on extending it continues.

Despite his criticism of DBMS products for their "*one size fits all*" approach, Stonebraker believed that relational technology was sound, particularly, SQL as a query language and in Jim Gray's transactions with *ACID* properties: *atomic*, *consistent*, *isolated* and *durable*. He was a critique of both NoSQL and MapReduce, and led a research group spread across MIT and other US universities that pioneered analytic databases building *C-Store* [13] a column store complying to ACID transactions. They too adopted MonetDB like vertical partitioning, but devised newer techniques for performance.

Both MonetDB and C-Store outperformed row store DBMSs and NoSQL products for analytic workloads, and popularized column stores for analytic SQL processing. There were several other industry products built on column stores, and today even the old school row DBMSs provide support for some form of column store, modern or simulated [14].

2.5 Knock! Knock! Who's There?

Around the same period of advances in SQL technology, hardware platforms and NoSQL, *cloud computing* through pay-by-use public services made distributed computing accessible to common folk. To use a high performing distributed computing system with assured QoS parameters, one need not be affiliated to a high profile research centre or corporate enterprise. The cloud made all of computing available through pay-by-use services, IaaS, PaaS, SaaS, AaaS, and eliminated the initial roadblock of high investment. And, public clouds provided nice and easy browser based GUIs for configuring distributed clusters.

Today, there are several public cloud vendors, Amazon, Microsoft, Alibaba, Internap, CenturyLink, etc., and private clouds may be built using virtualization services like VMware. Big Data processing shifted to public and private clouds. Capitalizing on advances in cloud technology that brought in easy scale-out opportunities to tackle data volume, several vendors of column stores adopted the MPP paradigm invented [5] by row stores to become parallel databases, and deployed on the cloud to tide past one of the 3Vs of Big Data: *volume*.

There were other benefits too: (1) cloud agnostic: deployable anywhere (2) elastic scale-out: no user intervention (3) cost savings: network storage, like Amazon EBS or Azure Premium IO, that decouples compute and storage; preserves data when compute nodes are shutdown; unlike attached storage, 24 × 7 is not mandatory (4) start small and scale-up: no high investment barrier.

But can SQL column databases do Big Data analytics? Back again at the Gates of Horn and Ivory of Big Data: Horn? Or Ivory?

2.6 Do Waves Turn into Tsunamis?

Database researchers have been critical of NoSQL and MapReduce. Stonebraker was one of the earliest to criticize MapReduce, a building block of NoSQL systems; he and his group compared [15] three approaches to data processing on a cluster of 100 nodes using a synthetic benchmark: MapReduce, a commercial row database and columnar Vertica. They found that both row and column DBs outperformed MapReduce, except that it was easier and faster to load data into MapReduce cluster: average 3.2x (row vs MR) and 7.4x (column vs MR).

Mohan, inventor of ARIES family of locking and recovery algorithms that have profoundly influenced DBMSs in guaranteeing ACID properties, is critical [16] of NoSQL systems for ignoring history, preferring expediency over rigor, failing to provide an easy query language and adopting ad-hoc solutions for inherently complex problems such as transactions, concurrency, standards, etc.

Twelve years later, Brewer the inventor of CAP theorem believes: "*2 of 3 formulation was misleading because it tended to over-simplify the tension among properties*" [17]. Advocates of NoSQL had given up consistency based on 2-of-3 limitation that was applicable only in the context of failures. All NoSQL products are not BASE: Neo4j is ACID, Bigtable is C+A.

More recently, in Big Data processing context, challenges with NoSQL and MapReduce [18] have been reported. NoSQL proponents have adopted SQL like Hive for query processing. And there is Pig, HaLoop, Spark, Mahout, ... no longer as simple as claimed to be; perhaps, getting to become as baroque as DBMSs!

The NoSQL criticisms of researchers couched in technicalities are neatly summed up in an informal style [19] addressing denormalization, consistency, absence of SQL, lack of standards, schema-less world, data movement cost and poor eco-system, ending with tongue-in-cheek lines: "*We tear things down only to build them back again... The king is dead. Long live the king.*".

It appears that SQL databases in parallel, columnar form are regaining lost ground, at least for structured Big Data. Perhaps, Darwin's natural selection applies to fluid dynamics and data processing too!

3 Column Stores

This section differentiates DSM and NSM, traces origins of column stores, and discusses architecture of MonetDB, C-Store, and some industry products.

3.1 What Is a Column Store?

Typically DBMSs store user data on persistent storage in fixed size blocks or pages, and all IO is performed at the level of pages. The primary difference between column and row stores lies in the way data is stored in the pages.

Pages store a header and rows of a table contiguously in some order, say, sequential. Within a row, data values of columns are stored contiguously and rows may also have a header. Size of rows in a page varies as some columns may be VARCHAR strings or values may be *null*. This type of row store is referred to as *N-ary Storage Model (NSM).* To retrieve 3rd column in 8th row of a page, the page containing 8th row has to be read, 8th row in the page accessed and 3rd column of row extracted. Even if a query uses only the 3rd column of a table, a sequential scan on row store retrieves data for all rows of the table.

Column stores perform a vertical partitioning of the table and store each column separately. Within a page, column values are contiguous across rows. When pages of a column are retrieved, data for only that column are accessed. For a query that needs only the 3rd column, sequential scan on column store retrieves data only for that column with bare minimum IO cost.

Generally, analytic query processing involves full table scans; most expensive operation is disk IO, even if system uses SSDs instead of spinning disks. As most queries access a few columns, out of wide table columns running to 10 s or 100 s, column stores gain significantly in performance by reducing IO cost.

Column stores have an additional benefit: type based, better compression. As a page stores data of same type for a column, information entropy of column is low with higher value locality; sorting data lowers entropy further. In contrast, NSM stores full rows with columns of different data types, and has high entropy. Compression of NSM pages is not type based, and has poorer ratios. High compression further reduces IO cost as table is smaller.

SQL queries work at the row level; if a query projects four columns of a table to generate 100 rows, the result is a matrix of size 100×4. For an equi-join on 5th column of tables $t1$ and $t2$ generating sum of columns 6 of $t1$ and 8 of $t2$, 6th and 8th columns of the two tables must be added only for rows that match on 5th column of both tables. In other words, *columns-of-row* property between columns of a table must always be preserved. This comes for free, and is easy in row stores but is not so easy in column stores. NSM may do better if all columns of a table of 100s of columns are retrieved by a sequential scan.

3.2 Early Origins

The original idea of organizing DB stores by vertical column partitioning was by Copeland and Khoshafian who proposed [20] the *Decomposition Storage Model (DSM)* in 1985. They used a surrogate key like row_id to preserve the columns-of-row property. Consequently, DSM storage sizes were higher than NSMs; also entropy increased due to row_id value, and there was no compression. DSM outperformed NSM only in some cases where just 1 or very few columns were selected. DSM went no further until MonetDB.

Well before MonetDB and C-Store, in 1996, Sybase released a product for analytics, SybaseIQ, using a column store. For DSS applications, [21] advocates major changes in DB store management; a few years later, the column store and bitmap indexes of SybaseIQ are described in [22]. SybaseIQ failed to disrupt the market, or even capture mind-share of academia or industry. It did not also take advantage of MPP. In the mid-nineties, there was no Web 2.0, Big Data, NoSQL, CPU caches or one-size-fits-all to catalyze a path change; or, Sybase was just plain unlucky: early bird missing worms that weren't yet ready!

3.3 Research Prototypes: MonetDB and C-Store

Architecture and internals of DSM stores MonetDB and C-Store are discussed.

MonetDB: A radical departure in architectural design of a DBMS, MonetDB invented novel techniques [12,23] for several phases of SQL processing. It used DSM storing each column of user table separately in a table called *Binary Association Table (BAT)*. A BAT was defined as a set of 2-ary tuples (*surrogate, value*) with the surrogate representing the row_id, also termed *OID*. The columns-of-row property is preserved by assigning the same OID for all column values of a table row. The OID values are not materialized in base BATs. Scalar types were stored in BAT by their values, while VARCHAR strings used a dictionary index into a heap that stored the strings.

The executor did not use the standard iterator model and planning was not by a cost based optimizer. Instead, it executed low level relational algebra called *BAT Algebra* which was optimized for column-at-a-time execution. BAT Algebra was executed by a virtual machine programmed in *MonetDB Assembly Language (MAL)*. The MAL program applied an operator to completion over its entire input data: a full column of a table, leading to materialization. Intermediate query results were preserved as BATs and *late materialization* techniques were adopted. Late materialization postpones stitching of tuples into rows, from individual columns, until late in query processing.

A key contribution of MonetDB was its hardware conscious query processing algorithms that took advantage of CPU cache [23,24]: *radix cluster* for hash join and *radix decluster* for SQL projection after a join and sorting. As MAL was generated for user queries, optimization was at runtime and done incrementally.

MonetDB used block-level processing at its extreme: the full column. It had no compression and without a buffer manager relied on OS for memory management. Also, it was not an MPP system.

Later enhancements to MonetDB targeted to eliminate DBMS' bells and whistles whose secrets were known only to a few of the high priests. Materialized views could be made automatic by a *Recycler* that cached and retrieved results from intermediate BATs of a query. Indexes could be maintained, reorganized and adapted, based on query workload, on-the-fly with *database cracking*.

The X100 project (Vectorwise) was hived off to pursue vectorization [25] to minimize effects of materialization. As an open source project, MonetDB is used [12] for emergency management, earth observation, astronomy, etc.

C-Store: Terming row stores of DBMSs as *write-optimized*, [13] proposes a *read-optimized* store with vertical partitioning of table columns for ad-hoc querying. Instead of storing each column separately, for performance reasons, they use *projections* (not SQL π), which store groups of user chosen columns in a specified sorted order. Projections may overlap, with storage duplication, and the planner picks the most appropriate projection for query processing.

Projection data is horizontally partitioned into *segments* based on data values of sort key of a projection; a segment is referred to by its *segment identifier*. To reconstruct rows of a table from segments containing partitions preserving the columns-of-row property of a table, *storage keys* and *join indexes* are used.

A storage key is just the physical position of the row in a segment, and is a mythical entity that is not physically stored in read-store. A join index is a collection of (segment identifier, storage key) pairs, an injective function between projections that may be interpreted as resorting projection $T1$ into the same order as another projection $T2$ with both $T1$ and $T2$ defined on table T.

Read-store data is stored in compressed form, using light-weight compression schemes that are not CPU intensive [26]: dictionary, run-length and bit-vector with null suppression. Executor works directly on compressed data deferring decompression to a later stage of query processing.

C-Store uses hybrid architecture for its data store: in addition to the read-optimized DSM, it implements a write-optimized store (WOS) to deal with infrequent updates and deletes that may occur in data warehouses. All data inserts and modifications are in WOS with delete bitmaps, and the system internally manages data movement between the stores with its *tuple mover*.

The planner converts a SQL query into a tree of C-Store query operators that is executed by an iterator model working at block-level of 64 K buffers. Though late materialization is not mentioned in [13], later publications [14,27] illustrate gains in C-Store from late materialization.

C-Store supports ACID transactions managed through MVCC like snapshot isolation, and its recovery uses ARIES like logical logging. Though, its architecture was designed for MPP, C-Store was not a parallel system. It does not support indexes on column store tables.

3.4 Industry Products

We discuss a few industry MPP products using DSM and deployed on cloud; the set is restricted to native column store products: not simulating [14] column stores. Essentially, products that came into existence after the research prototypes, and excludes products with column stores that are extensions of NSM.

Both MonetDB and C-Store led to industry products with researchers turning entrepreneurs; the X100 version of MonetDB with vectorization became Ingres Vectorwise and C-Store spawned Vertica. Vectorwise is not MPP.

Products from other vendors listed below support DSM stores, follow MPP architecture with horizontal scale-out by adding nodes. Comparison criteria are from [11] focusing on: DB store, clouds deployed on, cloud storage type, on-premise support and UDF availability as Big Data analytics in SQL uses UDFs.

Redshift, *Amazon*: Based on Paraccel, available only on AWS on preconfigured .xlarge/.8xlarge nodes using attached storage for DB; UDF: Python.

SQL Datawarehouse, *Microsoft*: Available only on Azure; uses blob (cheaper and slower) storage for DB store caching data on node attached storage during query runs; elastic scale out. UDF: PL/SQL type stored procedures in T-SQL.

Vertica, *HP/MicroFocus*: Product version of C-Store, available on AWS/Azure and recommends attached storage for DB. Also available as an appliance on vendor hardware; offers a customized version of R for data mining; UDF: C/C++.

dbX, *XtremeData*: Cloud agnostic: available on AWS, Azure and INAP; private clouds with VMware virtualization and commodity clusters. Can use both attached and network storage (EBS and Premium IO). UDF: C/C++, Python, PL/SQL like stored procedures in customized plpgSQL.

Greenplum, *EMC Pivotal*: Native row store supporting DSM with restrictions: append-only; AWS/Azure. Recommends attached storage for DB, but can access files stored on S3. Customized version of R and SQL interface to MADLIB. Also appliance on vendor configured hardware. UDF: C/C++, Python and plpgSQL.

Snowflake, *Snowflake*: Available only on AWS; uses S3 (cheaper and slower) storage for DB store caching data on node attached storage during query runs. No support for UDFs; elastic scale-out; vendor managed service.

We present architectural details of one of the industry products listed above: XtremeData's dbX.

- Analytics database with shared nothing, MPP architecture running on a cluster of nodes: a head node coupled to data nodes through a high-speed network; all nodes with their own compute, memory and storage resources.
- DB store is hybrid: a native column store with compression and a row store. Like MonetDB, each column of a table is stored separately with no physical storage of OID. Like C-Store, DSM pages are compressed; method is auto-computed for best compression ratio at load time for light weight (dictionary, runlength, packed runlength, FOR delta) and heavy weight (LZW) methods.
- Data distribution across cluster data nodes with DDL options: nearly equal round robin, hashed by one or more columns and single node placement. Range partitioning on column values for both column and row stores.
- Query execution model is not the tuple-level, iterator model. Planner generates a sequence of *macroQ* ops; each macroQ op is a set of 1 or more *microQ* ops that may be scheduled in parallel with block-level data flow between ops. Takes advantage of multi-core CPUs with threaded execution of microQ ops.
- Intra-query parallelism at two levels: across the cluster with parallel nodes running query ops on distributed data, and within a node microQ ops run as threads; block-level pipelining across microQ ops; parallel Linux aio for all IO; threaded node communication layer for compressed block-level data.
- Execution is micro optimized with thread-safe C code generation at runtime for all or part of microQ ops; C code is JIT compiled. Generated code uses query value specific, data dependent optimization and is targeted for modern CPUs to minimize cache stalls.

– Large memory of modern systems may be configured for in-memory cache for materialized intermediate data in compressed form and user data tables.
– Plans are optimized to minimize data movement in a MPP cluster including broadcast of small sized tables, runtime stats based refinement of join distribution methods, skew processing and dynamic partition pruning.
– ACID transactions with MVCC; optimized WAL for ARIES like recovery.
– B-tree indexes, local and global, primary/foreign key only on row stores; check and not-null constraints on both stores.
– High speed (several TBs/hr) parallel, bulk data load [28], and extract.
– Improved fault tolerance [29] in a heavily threaded, parallel environment.
– Unlike C-Store/Vertica no duplicate storage of columns, physical design of projections or sorting; simple DSM store model like MonetDB; no BAT.

4 Big Data Processing

Data analytics is usually preceded by steps of cleaning and loading of data.

Cleaning and Loading: Unlike some NoSQL products, SQL analytic databases require data to be *in-place* within DBMS before running queries, which may require preprocessing steps: data integration, cleansing and standardization. SQL data warehouses have addressed such problems through ETL tools [3] for preparing structured data. Tools address data quality issues and efficient loading of large volumes of data. Cleaning and loading of Big Data are no different [5]. Preprocessing may have more issues with NoSQL, as eco-system is poorer.

As shown in [15], loading data into databases is more expensive than into NoSQL tools. Analytic databases have improved their load rates through bulk loaders. Agile data loading through Linux aio, multithreading, bunched locking, minimal WAL logging and parallel COPY statement has been implemented [28] in dbX to achieve bulk data load rates in TBs/hr on both commodity hardware and cloud for its row store. dbX column store bulk load performance with compression surpasses its row store.

Data Analytics: Big Data analytics on structured data uses unsupervised or supervised data mining algorithms to build models that classify the data; subsequently the model may be used for scoring or prediction with unlabeled data. Data mining is a highly studied area with a variety of algorithms based on statistical techniques and mathematics.

Ordonez and his group have been investigating the suitability of SQL for implementing data mining problems over the years [30,31] and have published several mining algorithms in SQL: k-Means, Naive Bayes, Expectation Maximization (EM), dimensionality reduction through principal component analysis (PCA), n variables regression/correlation, etc. They use *sufficient statistics* discussed in [32] as a technique to decouple mining algorithms from data for better performance, and SQL integrated user defined functions (UDF) for iterative code. More recently, Ordonez concludes [33] that parallel columnar databases with support for UDFs can solve Big Data analytics problems.

Working with industry, MIT academics showed [34] that Vertica outperforms NoSQL graph databases for graph mining: PageRank, single source shortest path (SSSP) and HCC to find connected components. Apriori was always in SQL [35]. Efficient kNN algorithms using *Z-order* of *z-values*, mapping multidimensional data to one dimension by bit interleavings, are given [36] in SQL for kNN and kNN-joins. With mining algorithm implemented in SQL, both [34] and [36] extend queries with ad hoc relational filters on other columns with no extra cost.

Wu et al. [37] conducted a survey, IEEE KDD Top10, to rank data mining algorithms based on votes polled and citations. Addressing use of SQL for data mining, [11] tabulates a summary of Wu's survey results with details of several algorithms (k-Means, Apriori, Naive Bayes, EM, PageRank, kNN, C4.5/CART) of KDD Top10 implemented in SQL, and suggests that iterative nature of mining algorithms is a deterrent for SQL versions. Similar sentiments are echoed in MapReduce world too [18]. References for original algorithms in [37] and SQL versions in [11]; for brevity of space, only a few SQL version references here.

Security and Privacy: With proposed privacy regulations world over (e.g., EU's GDPR), a major issue confronting Big Data relates to security and privacy of data; legal impact of non-conformance is severe. Unlike NoSQL systems, databases have addressed the issue for decades: SQL roles, PAM access, data encryption, audit trails, etc. Full conformance may be easier in the SQL world.

5 Conclusion

On use of SQL databases for Big Data, Sam Madden, an architect of C-Store and co-author of [34], ends on a pragmatic note [38]: "*although databases don't solve all aspects of the Big Data problem, several tools – some based on databases – get part-way there*".

We agree, and hope that in the not far future, parallel, columnar databases will do Big Data in a *BIG* way! After all, metamorphosis in biology teaches us about the ugly caterpillar transforming itself into a beautiful butterfly!

References

1. Chamberlin, D.D., et al.: A history and evaluation of System R. Commun. ACM **24**(10), 632–646 (1981)
2. Graeffe, G.: Query evaluation techniques for large databases. ACM Comput. Surv. **25**(6), 73–170 (1993)
3. Chaudhuri, S., Dayal, U., Narasayya, V.: An overview of business intelligence technology. Commun. ACM **54**(8), 88–98 (2011)
4. Pavlo, A., Aslett, M.: What's really new with NewSQL? ACM SIGMOD Record **45**(2), 45–55 (2016)
5. Chen, M., Mao, S., Liu, Y.: Big data: a survey, mobile network applications. Mob. Netw. Appl. **19**, 171–209 (2014). Springer Science
6. Strauch, C.: NoSQL databases, selected topics on software-technology ultra-large scale sites. Stuttgart Media University, pp. 1–149 (2011). http://www.christof-strauch.de/nosqldbs.pdf

7. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. In: USENIX OSDI 2004, pp. 137–149 (2004)
8. Ailamaki, A., Dewitt, D.J., Hill, M.D., Wood, D.A.: DBMSs on a modern processor: where does time go? In: Proceedings of 25th VLDB (VLDB 1999), pp. 266–277 (1999)
9. Brewer, E.: Towards robust distributed systems. In: 19th ACM Symposium on Principles of Distributed Computing (PODC 2000), Portland, USA, pp. 7–10 (2000)
10. Stonebraker, M., Cetintemel, U.: One size fits all: an idea whose time has come and gone. In: IEEE International Conference on Data Engineering (ICDE 2005), pp. 2–11 (2005)
11. Sridhar, K.T.: Big data analytics using SQL: Quo Vadis? In: IFIP CONFENIS 2017, Shanghai, China, 13 p. (2017)
12. Idreos, S., et al.: MonetDB: two decades of research in column-oriented database architectures. IEEE Data Eng. Bull. **35**(1), 40–45 (2012)
13. Stonebraker, M., et al.: C-Store: a column oriented DBMS. In: Proceedings of Very Large Data Bases (VLDB 2005), Trundheim, Norway, pp. 553–564 (2005)
14. Abadi, D., Boncz, P., Harizopoulos, S., Idreos, S., Madden, S.: The design and implementation of modern column oriented database systems. Found. Trends Database **5**(3), 197–280 (2012)
15. Pavlo, A., et al.: A comparison of approaches to large scale data analysis. In: ACM SIGMOD 2009, Providence, USA, pp. 165–178 (2009)
16. Mohan, C.: History repeats itself: sensible and NonsenSQL aspects of the NoSQL hoopla. In: Proceedings of EDBT/ICDT 2013, Genoa, Italy, pp. 11–16 (2013)
17. Brewer, E.: CAP twelve years later: how the "rules" have changed. IEEE Comput. **45**(2), 23–29 (2012)
18. Grolinger, K., et al.: Challenges for MapReduce in big data. In: IEEE SERVICES 2014, Anchorage, USA, pp. 182–189 (2014)
19. Wayner, P.: 7 Hard truths about the NoSQL revolution. InfoWorld, July 2012
20. Copeland, G.P., Khoshafian, S.N.: A decomposition storage model. In: ACM SIGMOD 1985, Austin, USA, pp. 268–279 (1985)
21. French, C.D.: Teaching an OLTP database kernel advanced data warehousing techniques. In: IEEE International Conference on Data Engineering (ICDE 1997), pp. 194–198 (1997)
22. MacNicol, R., French, B.: Sybase IQ multiplex - designed for analytics. In: Proceedings of Very Large Data Bases (VLDB 2004), Toronto, Canada, pp. 1227–1230 (2004)
23. Boncz, P., Martin, L., Kersten, M.L., Manegold, S.: Breaking the memory wall in MonetDB. Commun. ACM **51**(12), 77–85 (2008)
24. Manegold, S., Kersten M.L., Boncz, P.: Database architecture evolution: mammals flourished long before dinosaurs became extinct. In: Proceedings of the VLDB Endowment (VLDB 2009), Lyon, France (2009). PVLDB **2**(2), 1648–1653
25. Boncz, P., Zukowski, M., Nes, N.: MonetDB/X100: hyper-pipeining query execution. In: ACM CIDR 2005, Asilomar, USA, 13 p. (2005)
26. Abadi, D.J., Madden, S.R., Ferreira, M.C.: Integrating compression and execution in column-oriented database systems. In: ACM SIGMOD 2006, Chicago, USA, pp. 671–682 (2006)
27. Abadi, D.J., Madden, S.R., Hachem, N.: Column-stores vs. row-stores: how different are they really? In: ACM SIGMOD 2008, Vancouver, Canada, pp. 967–980 (2008)

28. Sridhar, K.T., Sakkeer, M.A.: Optimizing database load and extract for big data era. In: Bhowmick, S.S., Dyreson, C.E., Jensen, C.S., Lee, M.L., Muliantara, A., Thalheim, B. (eds.) DASFAA 2014. LNCS, vol. 8422, pp. 503–512. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05813-9_34
29. Sridhar, K.T.: Reliability techniques for MPP SQL database product engineering. In: IEEE ICSRS 2017, Milan, Italy, 6 p., December 2017, to appear
30. Ordonez, C.: Programming the K-means clustering algorithm in SQL. In: AAAI KDD 2004, Seattle, USA, pp. 823–828 (2004)
31. Ordonez, C.: Statistical model computation with UDFs. IEEE Trans. Knowl. Eng. **22**(12), 1752–1765 (2010)
32. Graeffe, G., Fayyad, U., Chaudhuri, S.: On the efficient gathering of sufficient statistics from large SQL databases. In: AAAI KDD 1998, pp. 100–105 (1998)
33. Ordonez, C.: Can we analyze big data inside a DBMS? In: Proceedings of 16th International ACM Workshop on Data Warehousing and OLAP (DOLAP 2013), pp. 85–92 (2013)
34. Jindal, A., Madden, S., Castellanos, M., Hsu, M.: Graph analytics using the Vertica relational database. In: IEEE Big Data, Santa Clara, USA, pp. 1191–1200 (2015)
35. Sarawagi, S., Thomas, S., Agrawal, R.: Integrating association rule mining with relational database systems: alternatives and implications. In: ACM SIGMOD 1998, Seattle, USA, pp. 343–354 (1998)
36. Yao, B., Li, F., Kumar, P.: K nearest neighbor queries and kNN-joins in large relational databases (almost) for free. In: IEEE ICDE 2010, pp. 4–15 (2010)
37. Wu, X., et al.: Top 10 algorithms in data mining. Knowl. Inf. Syst. **14**, 1–37 (2008). Springer
38. Madden, S.: From databases to big data. IEEE Internet Comput. **16**(3), 4–6 (2012)

Mining of Massive Datasets

Improving Infrastructure for Transportation Systems Using Clustering

Prashant Rajput, Durga Toshniwal, and Apeksha Agggarwal(✉)

Indian Institute of Technology, Roorkee, Roorkee, India
prashantrajput1992@gmail.com, durgatoshniwal@gmail.com,
apeksha.aggarwal785@gmail.com

Abstract. Transportation systems are called the lifeline of any urban area. Major transportation systems includes cars, taxis, buses, trams etc., which carries most of the local transport in a city. This work encourages the more use of public transport over private vehicles so as to save environment, energy and resources by suggesting improvement in infrastructure of bus services. Experimental work on data collected from city of New York is presented in this work. This data collected from taxis is used to analyze the presence of traffic in the area. Temporal data segmentation with respect to different time zones is performed considering the dynamic patterns of urban traffic. Next, popular data mining techniques of clustering are applied on this segmented data to form clusters for each time zone so as to identify areas of high traffic. Further, another data set of bus stops is used to identify places with no bus stops and high traffic congestions. Henceforth new bus stops are suggested on places with high traffic density and no bus stops. Thus, a comparative study over baseline is done to recommend places that require bus stops.

Keywords: Bus stop · Traffic · Pick-up · Latitude · Longitude

1 Introduction

Urbanization has led to formation of smart cities which in turn has improved the lives of people living in the cities, but is also the origin of varied problems such as pollution, traffic congestion, energy consumption etc. Today, urban cities infrastructure generates large amount of data in smart cities. This data can help us solve the problems if the data is used wisely, including problems of transportation. One of the major problem urbanization has caused is the traffic congestion, which is to be addressed for smart transportation [1] of people and goods. Traffic congestion problem further leads to wastage of scarce resource of human productive hours along with the degradation of environment in terms of fuel wastage and environmental pollution.

Owing to the adversities due to traffic most of the countries [2] around the world are taking productive steps for human health and environment by encouraging the use of public transport rather than the use of private diesel cars. Public

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 129–143, 2017.
https://doi.org/10.1007/978-3-319-72413-3_9

transport system includes bus, metro, trams, ferries etc. The motivation for using the data mining techniques [1] in smart transportation system is to handle the resources such that it improves the environment as well as the living standards of people of the cities in a positive manner. Bus transportation system is thus the need of the hour which is focused mainly in this work.

In the past many problems related to smart transportation or smart mobility have been addressed. First of them is related to improving experiences of driver traversing over the roads: Fast driving routes generally save time as well as fuel because on such routes traffic congestion is less. In the past a lot of studies have been performed to determine real-time traffic situation, learn traffic patterns and forecast traffic. Thiagarajan et al. [3] suggested VTrack, a hidden markov model (HMM) based technique to calculate the travel time and to suggest the paths most likely to be visited by people. T-Drive [4] is a system that provide driving directions according to traffic, driving habits of drivers, weather conditions etc. This system also recommends optimal paths to drivers to traverse on empty roads. Wang et al. [5] proposed a model which is real time and city wide for estimating travelling time on any road segments using trajectory data. There were a few challenges considered in this work. Firstly, lack of sufficient amount of data over varied road segments and different time slots. Secondly, there are a number of ways to combine the trajectories for estimating travel time. Finding the best combination of trajectory is a complex task. Thirdly, real time updating of user query generating dynamically in the system anywhere, anytime is further a problem. Wang et al. [5] proposed to use different drivers' travel time on different segments at different time periods which is a combined form of historical, space related and time related context from the trajectories. This method performed better than the baseline approaches.

Taxis are an important part of both public and private transportation system of a city for providing door to door services. In cities people wait for non-trivial time for empty taxis while at other end cabs drivers wait deliberately to get commuters. Efficiently connecting vacant taxis with waiting passengers is important for saving time of the commuters, reducing energy consumption, reducing traffic, increasing profits etc. The main issue in this system was the uncertainty of the movement of the taxi while searching for the taxi and managing them is a further induced problem to be addressed. Traffic conditions should also be taken into consideration while estimating pick-up time of the user. The algorithms used for finding free cabs are flooding and probabilistic [6]. In flooding routing algorithm client request is broadcasted to all neighbors recursively until request reaches the free cab or TTL (time to live) becomes zero. To prevent loops, visited nodes are marked. In probabilistic algorithms, routing tables are built for all the nodes within TTL hop limit. Entries consists of probability of finding a free taxi which is updated using neighbors' node. Ge et al. [7] developed a recommendation system, to recommend the pick-up points and parking lots to the taxi drivers. This system aims at minimizing the energy wastage and maximizing profit of taxis. T-Finder [8] system gives the taxi drivers routes to the location where chances of getting passengers are high. This system also suggests nearby location points where empty cabs may be found out.

Bus transportation is very important public transport system on which most people rely for commuting. In order to improve the bus transportation system, it is required to make bus services reliable and more frequent. Watkins et al. [9] studied the influence of the giving real-time information about bus arrival, not only reducing the waiting time of people waiting at bus-stops but also of people who plan their journey on basis of such information. In case if GPS systems are not installed on the buses then alternative solution was given in [10]. Zimmerman et al. [10] proposed to use the GPS traces of the commuters of the buses. Then collected traces were processed and real-time predictions were made. Bastani et al. [11] proposed a system called flexi in which bus routes were found out by analyzing the trip data of very large set of the taxis.

Our proposed research work is different from previous approaches in the following aspects. The systems that have developed so far either focuses on improving driving experiences of drivers by estimating time, providing directions to destination or on improving search experience of taxis or bus services. But there is not any significant work done on finding out the new places in the cities where large number of people commute, using the GPS trajectories of large set of taxis and then checking whether bus routes are provided to that place or not. Clustering algorithms have been used in our work to solve this problem. Density based clustering algorithms are of $O(n^2)$ time complexity thus these are not scalable. Partitioning based clustering algorithms takes $O(n)$ time, where n is the number of data points. Among partitioning based clustering algorithms, k-means and k-means++ are employed in this work. In k-means clustering, initial selection of k points is random and this may lead to bad clusters. Thus, k-means++ clustering tries to remove this limitation of k-means.

Section 2 discusses the proposed framework in detail for solving the specified problem. Section 3 discusses the dataset and implementation. Section 4 discusses results in great detail and Sect. 5 presents conclusion and future work.

2 Proposed Framework

Proposed framework to solve the identified problem is discussed in this section.

2.1 Problem Statement

Threefold objectives of this work are given further in order to improve the bus transportation:

Problem 1. First is to find out the interesting patterns in the traffic volume data of taxis and generating heat maps to identify the areas with more number of commuters in the city.

Problem 2. Second is to find out the most traffic congested areas and among these areas select areas that do not have bus stops around them.

Problem 3. Third is to provide recommendations about areas where new bus stops needs to be established.

Algorithms for suggesting bus stops is discussed in Sect. 2 in detail. Solution to these three objectives have been decomposed in following sub tasks:

1. Data Preprocessing and Cleaning: In this step, missing values are treated and taxi data is transformed to preprocessed data.
2. Temporal Data Segmentation: Dividing data into time zones as traffic in the cities is variable with respect to time.
3. Geocoding: Geocoding of existing bus stops with respect to their actual locations.
4. Detecting pick-up points: Finding areas in the city contributing to high pick-ups by taxis.
5. Recommendation: Finally recommendations for new bus stops is given.

2.2 Data Mining

Steps for data mining on taxi dataset are shown in Fig. 1.

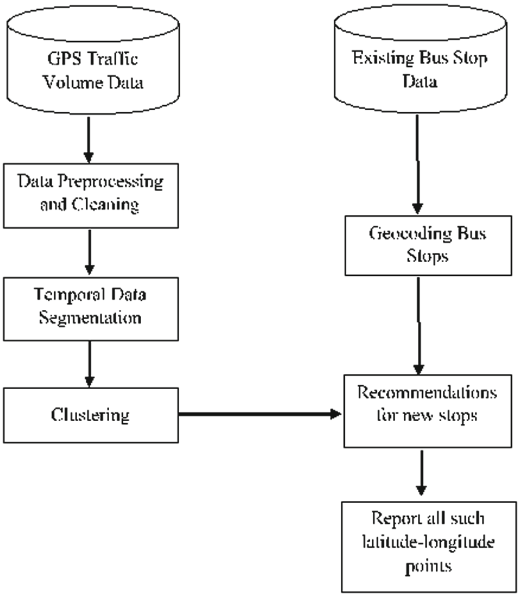

GPS Traffic Volume Data
Data Preprocessing and Cleaning
Temporal Data Segmentation
Clustering
Existing Bus Stop Data
Geocoding Bus Stops
Recommendations for new stops
Report all such latitude-longitude points

Fig. 1. Steps of data mining on taxi dataset

Data Preprocessing and Cleaning. In the data preprocessing step data cleaning is done. Taxi data contained some irregularities such as some attribute values were missing. Such records have been deleted as missing data records were very few as compared to the amount of dataset used. The necessarily considered and selected in this work are the pick-up latitude, pick-up longitude, drop-off latitude and drop-off longitude, pick-up time.

Traffic Analysis and Data Segmentation. Traffic pattern keeps on changing. It is different during morning hours, during evening hours, during afternoon and during night. Some places have high traffic during morning such as industrial areas, colleges etc. and some areas have high traffic during evening such as malls, clubs etc. So, the whole data have been divided into 4 time zones each of 6 h as shown in Table 1.

Table 1. Temporal data segmentation

Time zone number	Time interval
Time zone 1	0:00 to 5:59 h
Time zone 2	6:00 to 11:59 h
Time zone 3	12:00 to 17:59 h
Time zone 4	18:00 to 23:59 h

To show that the traffic varies in these four time zones, heat maps have been used in this work. Heat maps are a great tool for representation of the data. They show the intensity of the data plotted on geographical maps. In this work heat maps are superimposed on Google maps that shows the traffic intensity on different places.

Clustering. Aim of the clustering is to increase the inter-cluster distance and minimize the intra-cluster distance. Clustering algorithms are used to divide the data points and arrange them into a number of groups known as clusters. Following clustering algorithms have been applied.

1. K-means Clustering: In k-means clustering [12] centroid of cluster is used to represent the cluster. Centroid of cluster is mean of data points within the cluster. K-means clustering randomly selects k data points each of which is cluster center. For remaining points, each point is assigned to one of the cluster from where that point is closest on the basis of Euclidean distance. Time complexity is *O(nkt)* where n is total data points, k is total clusters and t is number of iterations.
2. DBSCAN: Density based spatial clustering of applications with noise (DBSCAN) [12] is a density based clustering algorithm. Initially all points are marked as unvisited. Algorithm selects one of the point p, if p is unvisited it marks p as visited, then it checks if ε neighbor of p has at least minimum number of points. If p satisfies the criteria it is assigned to a new Cluster C, otherwise it is marked as noise. All those points which are in e neighbor are added to N. Algorithm iteratively joins points to C in N that are not added to any cluster. Cluster C is expanded until it can't be further expanded. To find next cluster, algorithm selects the unvisited point and repeats same process. Time complexity of DBSCAN is $O(n^2)$ where n is total data points.

3. K-means++ Clustering: K-means clustering [13] have some limitations. One of the limitation is that in k-means clustering initial selection of data points is random, so sometimes final clusters formed are dependent upon the initial selection of k centers. This limitation is removed by k-means++ clustering. In this clustering, initial selection of data points is not random. They are selected intelligently. Let the smallest distance between initial cluster center and any other data point be D_i. Let there are n data points $x_1, x_2, ...x_n$. In k-means++ clustering number of clusters must be decided beforehand. Approximate number of clusters, k is decided by (1).

$$k = \sqrt{\left(\frac{n}{2}\right)}. \tag{1}$$

Where k is number of clusters and n is the number of records. Clustering algorithms are applied on the dataset of each of the four time zones on the basis of pick-up points. Results of the clustering will give the areas around which large number of people commute. These areas will have high pick-up and drop-off points around them.

Equation (1) gives the rough estimate of the number of clusters so it may be possible that a large cluster is divided into number of sub clusters. So the task is to find such clusters and merge them and calculate new centroid to represent the merged clusters. To find such clusters distance matrix is created. Distance matrix shows that if distance of a cluster a with cluster b is less than 400 meters than corresponding entry in the distance matrix is marked as '1', otherwise entry is marked as '0'. These marking in distance matrix helps us in identifying the clusters that need to be merged. Clusters are merged repeatedly until no more merging can be performed. Algorithm 1 depicts the algorithm for merging such clusters.

Geocoding. Data of bus stops is in the form of postal addresses. Hence such data are required to be converted to respective latitude and longitude points. Address must be given in detail to fetch correct latitude-longitude point. If address is not given in the detailed form, latitude-longitude point we get is approximate.

Scikit−learn tool provided by Python has been used for geocoding [14]. Tool provides library that accurately converts addresses to the latitude-longitude point [15] in order to fetch bus stops in the form of latitude, longitude points. Method that is used for finding distance between two latitude-longitude points is Vincenty's formula. Vincenty's formula [19] is more accurate than the Great-circle distance because Vincenty's assumes that earth is oblate spheroid whereas Great circle distance assumes earth is spherical.

After merging the clusters, new clusters are represented as point (latitude, longitude) by the means of latitudes and longitudes of all those cluster centers' which are merged with respect to Eqs. (2) and (3).

$$\text{Latitude of new merged cluster} = \frac{\sum latitude}{m}. \tag{2}$$

$$\text{Longitude of new merged cluster} = \frac{\sum longitude}{m}. \tag{3}$$

```
Input: Set of cluster centers representing clusters
Output: Set of modified cluster centers so that none of them are distance
        400 m or less apart
Algorithm algo1()
    Call calculate procedure distance_matrix on initial set of clusters;
    repeat
        Call merge_clusters procedure;
        Call distance_matrix procedure on newly formed clusters;
        Exit if all the entries in distance matrix are 0;
    until all entries in distance matrix are 0;
Procedure distance_matrix()
    Take an empty 2D array representing cluster centers on both sides;
    Traverse all the cluster center and check;
    if distance between 2 cluster is v less than 400 $m$ then
        mark an entry 1 in corresponding place in 2D array
    else
        mark entry 0 in corresponding place in 2D array
    end
    Return the distance matrix;
    return;
Procedure merge_clusters()
    Take an empty cluster list;
    repeat
        Traverse the distance matrix and if entry is 1, append cluster center to
          empty cluster list;
        Delete old merged clusters and add new clusters after merging to the
          clusters list;
    until all clusters not get merged;
    Return new set of clusters in the list .;
    return;
```

Algorithm 1. Formation of clusters of pick-up and drop-off points

Finding New Stops. Proposed algorithm for finding new bus stops are discussed in this section. According to Jarrett et al. [16], distance between two bus stops in a metro city must be near mile i.e. 400 m. This is the reason why in above code distance 400 m is used to find out new bus stops. Given pseudo code in Algorithm 2 tries to find out new bus stops by finding out which cluster center is not covered by any of the existing bus stops i.e. which cluster center is not within 400 m distance of any existing bus stops.

```
Input: Set of all cluster centers representing clusters and all bus stops
Output: Set of cluster centers recommended
Traverse cluster centers;
Traverse bus stops;
If cluster center is not close to any bus stop then recommend cluster center for
  new stop;
```

Algorithm 2. Algorithm for recommending bus stops

3 Dataset and Implementation

Primarily two datasets have been used in this work. First data is taxi volume data of New York City, USA for the month of June 2016 [17]. This data contains 18 attributes. Some of the attributes are pickup_datetime, dropoff_datetime, trip_distance, pickup_longitude, pickup_latitude etc. Out of these attributes that are important and has been used in this work are pickup_datetime, dropoff_datetime, pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude. Total number of records in taxi volume data is 62,49,304 [17]. Second dataset is of addresses of the existing bus stops in the New York City, USA. As bus stops were given in the form of postal addresses, they needed to be converted into latitude-longitude points for doing computations as discussed in Sect. 2 [18].

3.1 Data Segmentation

Whole data has been divided into 4 time zones each of 6 h. Table 2 shows the number of records in each time zone. Figure 2 shows the variation of number of pick-ups by the taxis each hour of the day. Figure shows that during morning hours there is much less traffic compared to other times of the day. Pick-ups of passengers increases as day passes till 18:00 h. In the evening, the number of pick-ups decreases slightly. Again, during night traffic increases. Hence the reason for division into four time zones, each of six hour duration respectively.

Table 2. Number of records in each time zone

Time zone number	Number of records
Time zone 1	11,33,806
Time zone 2	14,61,442
Time zone 3	16,03,335
Time zone 4	17,72,738

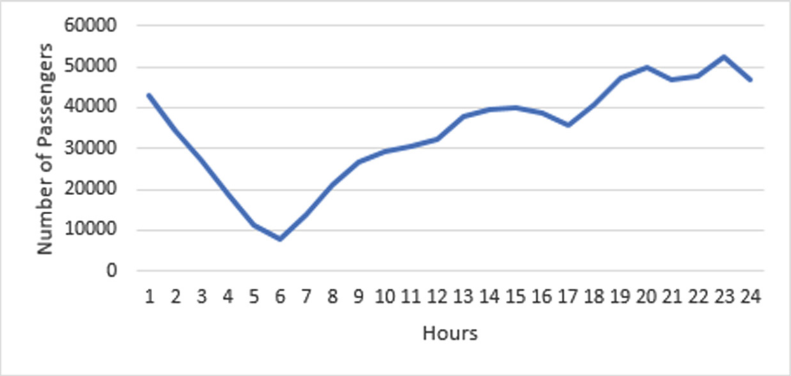


Fig. 2. Variation of number of passengers in each hour

3.2 Heat Maps

Figure 3 show the four heat maps superimposed on Google maps depicting data of different time zones. We can see different patterns of pick-up and drop-off points of the taxis in each time zone interval of 6 h. Areas with high traffic are marked with red color, normal traffic is marked with yellow and areas with comparatively less traffic are marked with green color.

Reddish-yellow point on the Fig. 3a shows the point on the map where number of people picking up or dropping off taxis are relatively larger than the greenish areas on the map. In time zone 1 i.e. between 0:00 to 5:59 h large number of people commuting are around Lower Manhattan and Midtown. But there is comparatively less number of pick-up and drop-off on Upper-East Side and Upper-West Side. Figure 3b shows that density of pick-up and drop-off point is relatively same in whole Manhattan with slightly high density in Midtown, Upper East Side and Upper West Side. Figure 3c shows that density of pickup and drop-off points is relatively higher in Midtown, Upper West Side and Upper East Side between 12:00 to 17:59 h. Figure 3d shows that between 18:00 to 23:59 h, there is pickup and drop-off largely only in the areas around Midtown and Lower Manhattan. Four heat maps shown in Fig. 3 depicts that pattern of pick-up and drop-off points of the taxis varies in 4 time zones of 6 h each.

4 Results and Discussions

Heat maps generated in Sect. 3 have shown the variation in transportation at different time zones. Recommendation for bus stops using clusters for different time zones, generated from taxi data are discussed in this section.

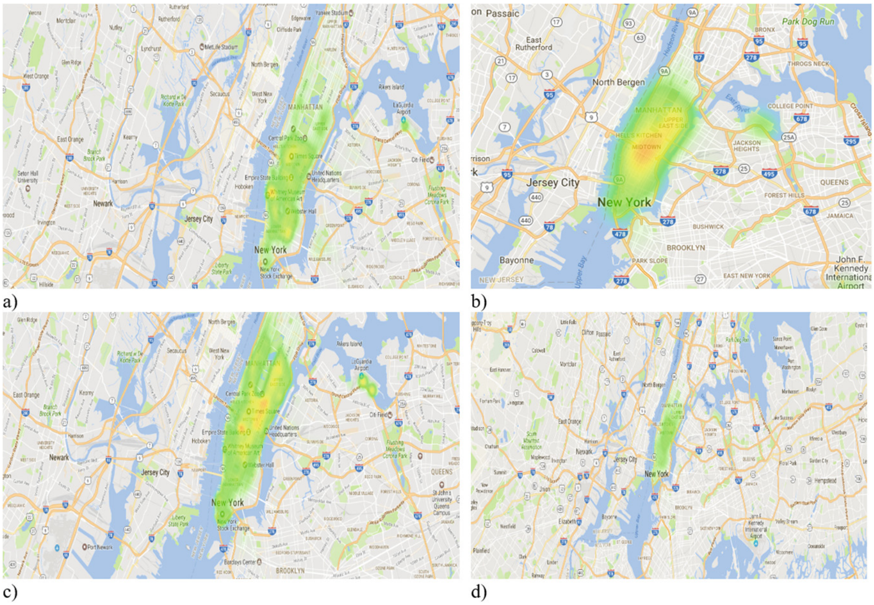


Fig. 3. Heat map superimposed on Google-maps showing density of pick-up and drop-off points between (a) 0:00 to 5:59 h (b) 6:00 to 11:59 h (c) 12:00 to 17:59 h and (d) 18:00 to 23:59 h (Color figure online)

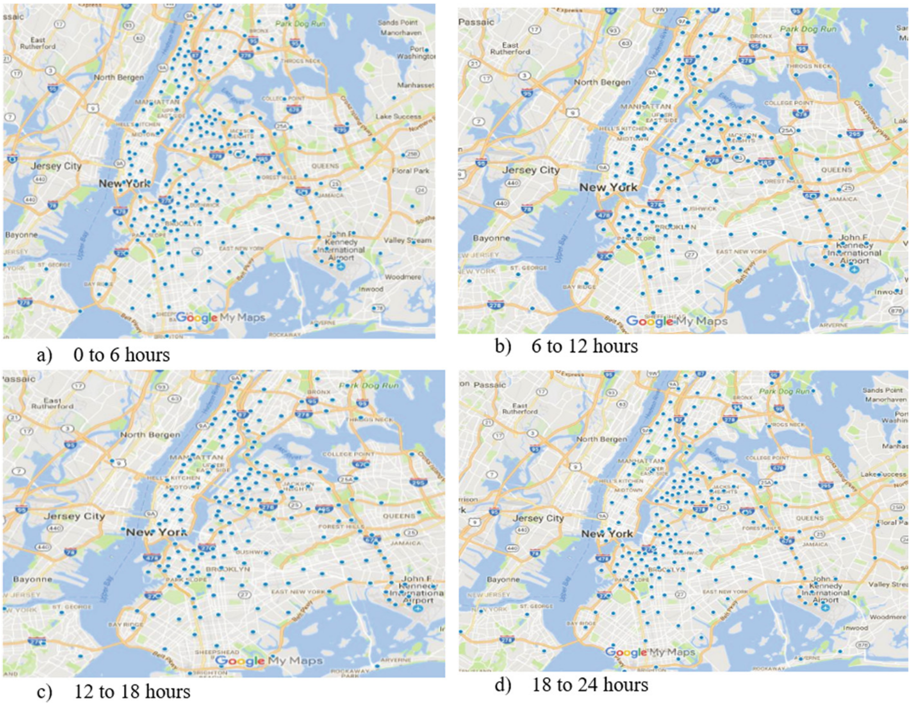
4.1 Clustering

Table 3 shows number of clusters before and after applying Algorithm 1. Results of applying k-means++ clustering algorithm over four time zones is shown in Fig. 4.

Table 3. Number of clusters in each time zone

Time zone number	Number of clusters	Number of clusters after merging
Time zone 1	816	413
Time zone 2	854	384
Time zone 3	895	358
Time zone 4	941	346

As we can see in Fig. 4 different time zones depict different patterns of intensities of pick-up and drop-off points of taxis. In Fig. 4a, blue dots depict the cluster of k-means++ clustering in time zone 1. These dots show that the areas around these blue dots have high pick-up and drop-off of passengers by private taxis. Similarly Fig. 4a–d depicts four time zones with clustered dots.



a) 0 to 6 hours

b) 6 to 12 hours

c) 12 to 18 hours

d) 18 to 24 hours

Fig. 4. Google map depicting centers of k-means++ clustering over data of different time zones i.e. between time: (a) 0:00 to 5:59 h (b) 6:00 to 11:59 h (c) 12:00 to 17:59 h and (d) 18:00 to 23:59 h (Color figure online)

4.2 Recommendation

Since bus stops are given in form of postal addresses, they needed to be converted into latitude longitude form. In order to make sure that area around bus stops have generally high traffic and high number of pick-ups by taxis, sample of taxi dataset have been taken. Clustering is applied on this sample dataset. Total number of clusters formed are 443. These clusters are within the 400 meters distance from the nearest bus stop. Error percentage is calculated using (4), where n is number of clusters in sample data and N is total number of bus stops.

$$\text{Error in } \% = \frac{n - N}{N} * 100. \tag{4}$$

Error came out to be 11.5% which is quite less. To find the new bus stops, all those cluster centers have been recommended as new bus stops which are not within 400-meter radius of any bus stop. In other words, all those areas which are not covered by any existing bus stops. Latitude and Longitude of areas where bus stop need to be build are shown in Table 4. Recommended bus stops are shown in Fig. 5.

Table 4. Latitude and longitude of new recommended stops

Stop number	Latitude	Longitude
1	40.6171	74.0337
2	40.6934	−73.8543
3	40.6211	−73.9326
4	40.6076	−74.0774
5	40.7317	−73.7782
6	40.5872	−73.8142
7	40.6315	−73.9874

Three clustering algorithms have been applied on the data set namely k-means++, k-means and DBSCAN. Out of these clustering k-means++ algorithm found out 7 new clusters which requires new stops but k-means found out only 2 bus stops and these two stops are already been covered by k-means++. DBSCAN couldn't found out any new stop. This is depicted in Table 5. Thus, among these clustering algorithm K-means++ gave the best result. The reason why DBSCAN did not performed well is that the density of the data points in the dataset is quite large, so DBSCAN keeps on expanding its cluster and thus merges cluster that does not have a bus stop to the cluster that already have a bus stop.

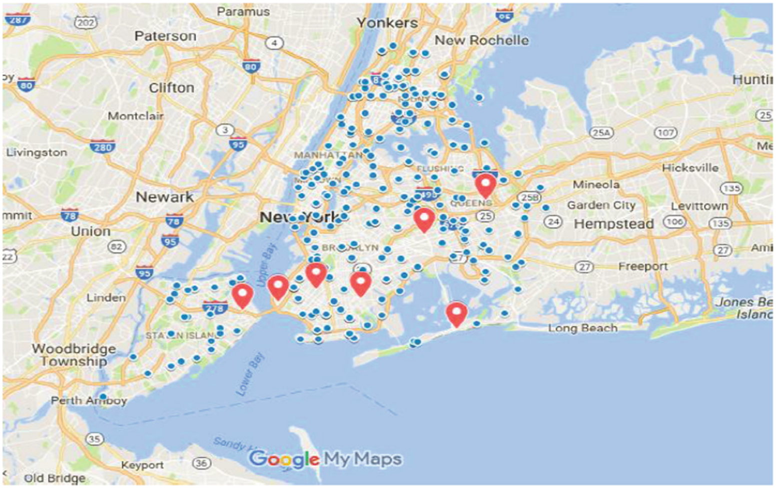


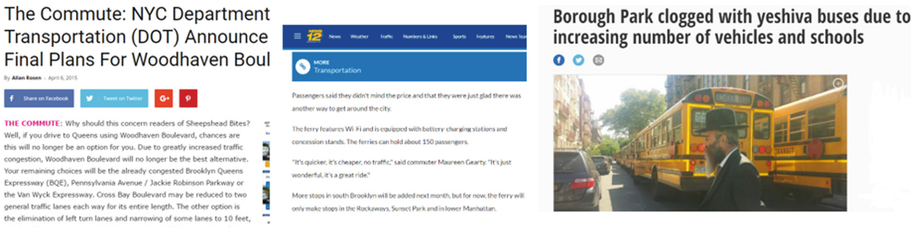
Fig. 5. Google map showing existing bus stops (blue points) and recommended bus stops (red markers) (Color figure online)

Table 5. Comparison between results of three clustering algorithms

Algorithm:	K-means++				K-means				DBSCAN			
Tz	TZ_1	TZ_2	TZ_3	TZ_4	TZ_1	TZ_2	TZ_3	TZ_4	TZ_1	TZ_2	TZ_3	TZ_4
No of clusters	413	384	358	346	621	529	573	594	205	219	212	231
No of new bus stops found	7				2				0			

4.3 Comparison with Ground Truth

Figure 6a–c shows the ground truth. Figure 6a shows that New York Department of Transportation is searching for new plans to reduce traffic in Woodhaven Boulevard (Stop number 2 in Table 4. Figure 6c shows congestion in Borough Park (Stop number 7 in Table 4) due to opening of new schools.



The Commute: NYC Department Transportation (DOT) Announce Final Plans For Woodhaven Boul

By Allan Rosen - April 6, 2015

THE COMMUTE: Why should this concern readers of Sheepshead Bites? Well, if you drive to Queens using Woodhaven Boulevard, chances are this will no longer be an option for you. Due to greatly increased traffic congestion, Woodhaven Boulevard will no longer be the best alternative. Your remaining choices will be the already congested Brooklyn Queens Expressway (BQE), Pennsylvania Avenue / Jackie Robinson Parkway or the Van Wyck Expressway. Cross Bay Boulevard may be reduced to two general traffic lanes each way for its entire length. The other option is the elimination of left turn lanes and narrowing of some lanes to 10 feet, which will have to accommodate large trucks. Will this even be safe?

MORE Transportation

Passengers said they didn't mind the price and that they were just glad there was another way to get around the city.

The ferry features Wi-Fi and is equipped with battery-charging stations and concession stands. The ferries can hold about 150 passengers.

"It's quicker, it's cheaper, no traffic," said commuter Maureen Gearty. "It's just wonderful, it's a great ride."

More stops in south Brooklyn will be added next month, but for now, the ferry will only make stops in the Rockaways, Sunset Park and in lower Manhattan.

Borough Park clogged with yeshiva buses due to increasing number of vehicles and schools

a) b) c)

Fig. 6. (a) Traffic congestion in Woodhaven. (b) New ferry service in Brooklyn. (c) Traffic congestion in Borough Park.

5 Conclusion and Future Work

Smart transportation problems includes traffic issues, parking issues, faster and smooth commutation of people and many others, which in turn affects environment as well as productive human hours. Witnessing to huge traffic problems via private vehicles, more use of public transport is the need of the hour. Among prominent of smart transportation problems, this work focuses upon improvement of bus services infrastructure by identifying traffic and suggesting places for new bus stops. Data collected of New York City is of GPS traces of taxis and of existing bus stops. In this work, clustering algorithms are used on the basis of pick-up points of GPS traffic volume data to find out places having high number of pick-ups by taxis suggesting high traffic in that area. All those areas which are identified by clusters and are not having bus stops in their neighborhood are recommended for making new bus stops.

In the data preprocessing step records containing missing values are removed. After data preprocessing step number of records reduced from 62,49,304 to 59,71,321. In temporal data segmentation step records are divided into 4 time zones of 6 h each because traffic pattern keeps on changing each hour. Thus, clustering without considering time will not be of any significance. Results of clustering algorithm gave the areas around which large number of commute and large number of taxis pick-up passengers from those places. Clusters not having a bus stops in their close neighborhood are recommended for new stops. Total number of new bus stops that are recommended are 7. These 7 areas have high number of pick-ups by taxis and doesn't have bus stops nearby.

Further there is some future scope for proposed research work. In this work number of passengers assumed per taxi is only one. However in real life situations, if more number of passengers are traveling in a taxi, this would generate more traffic counts for a particular area. Considering more number of people in a taxi, results for the same may be improved. This could be the future scope of this work. Further, most of the metropolitan cities behaves the similar way. Hence proposed system can be applied to other cities as well as per the availability of data set.

References

1. Aggarwal, A., Toshniwal, D.: Data mining techniques for smart mobility - a survey. In: Proceedings of the 5th International Conference on Advanced Computing, Networking, and Informatics. Springer, Goa (2017)
2. Business Insider. http://www.businessinsider.in/tech/10-cities-that-are-starting-to-go-car-free/slidelist/53726274.cms
3. Thiagarajan, A., Ravindranath, L., LaCurts, K., Madden, S., Balakrishnan, H., Toledo, S., Eriksson, J.: VTrack: accurate, energy-aware road traffic delay estimation using mobile phones. In: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, pp. 85–98. ACM (2009)
4. Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., Huang, Y.: T-Drive: driving directions based on taxi trajectories. In: Proceedings of ACM SIGSPATIAL Conference on Advances in Geographical Information Systems, pp. 99–108. ACM (2014)
5. Wang, Y., Zheng, Y., Xue, Y.: Travel time estimation of a path using sparse trajectories. In: Proceedings of the 20th SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 25–34. ACM (2014)
6. Zhou, P., Nadeem, T., Kang, P., Borcea, C., Iftode, L.: EZCab: a cab booking application using short-range wireless communication. In: Pervasive Computing and Communications, pp. 27–38. IEEE (2005)
7. Ge, Y., Xiong, H., Tuzhilin, A., Xiao, K., Gruteser, M., Pazzani, M.: An energy-efficient mobile recommender system. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 899–908. ACM (2010)
8. Yuan, N.J., Zheng, Y., Zhang, L., Xie, X.: T-finder: a recommender system for finding passengers and vacant taxis. Trans. Knowl. Data Eng. **25**, 2390–2403 (2013)
9. Watkins, K.E., Ferris, B., Borning, A., Rutherford, G.S., Layton, D.: Where is my bus? Impact of mobile real-time information on the perceived and actual wait time of transit riders. Transp. Res. Part A Policy Pract. **45**, 839–848 (2011)
10. Zimmerman, J., Tomasic, A., Garrod, C., Yoo, D., Hiruncharoenvate, C., Aziz, R., Steinfeld, A.: Field trial of Tiramisu: crowd-sourcing bus arrival times to spur co-design. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1677–1686. ACM (2011)

11. Bastani, F., Huang, Y., Xie, X., Powell, J.W.: A greener transportation mode: flexible routes discovery from GPS trajectory data. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 405–408. ACM (2011)
12. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques, 3rd edn. Elsevier, Waltham (2011)
13. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027–1035. Society for Industrial and Applied Mathematics (2007)
14. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
15. Python. https://pypi.python.org/pypi/geopy
16. Walker, J.: Purpose-driven public transport: creating a clear conversation about public transport goals. J. Transp. Geogr. **16**, 436–442 (2008)
17. NYC Taxi and Limousine Commission-Trip Record Data. http://www.nyc.gov//html/tlc/about/trip_record_data.html
18. Metropolitan Transportation Authority. http://bustime.mta.info/routes
19. Vincenty, T.: TDirect and inverse solutions of geodesics on the ellipsoid with application of nested equations. Surv. Rev. **23**, 88–93 (1975)

Temporal Topic Modeling of Scholarly Publications for Future Trend Forecasting

Amol P. Bhopale(✉) and Sowmya Kamath Shevgoor

Department of Information Technology,
National Institute of Technology Karnataka, Surathkal, India
amolpbhopale@gmail.com, sowmyakamath@nitk.edu.in

Abstract. The volume of scholarly articles published every year has grown exponentially over the years. With these growths in both core and interdisciplinary areas of research, analyzing interesting research trends can be helpful for new researchers and organizations geared towards collaborative work. Existing approaches used unsupervised learning methods such as clustering to group articles with similar characteristics for topic discovery, with low accuracy. Efficient and fast topic discovery models and future trend forecasters can be helpful in building intelligent applications like recommender systems for scholarly articles. In this paper, a novel approach to automatically discover topics (latent factors) from a large set of text documents using association rule mining on frequent itemsets is proposed. Temporal correlation analysis is used for finding the correlation between a set of topics, for improved prediction. To predict the popularity of a topic in the near future, time series analysis based on a set of topic vectors is performed. For experimental validation of the proposed approach, a dataset composed of 17 years worth of computer science scholarly articles, published through standard IEEE conferences was used, and the proposed approach achieved meaningful results.

Keywords: Machine learning · Data analytics
Frequent pattern mining · Topic discovery · Trend forecasting

1 Introduction

Deriving topics from a large set of document corpus and predicting their popularity is a critical requirement in personalized applications like recommender systems, e-commerce, e-learning etc. As per recent statistics, thousands of documents are being continuously indexed on the Web each minute of the day, which is a testimony to its ever-increasing popularity. The problem of effectively processing, storing and retrieving documents as per a defined user need is further compounded due to the wide diversity of the corpus, which means incremental approaches that are capable of dealing with this are needed. Managing these documents for making best use of them across domains becomes a critical problem

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 144–163, 2017.
https://doi.org/10.1007/978-3-319-72413-3_10

for Information Retrieval/Extraction (IR/IE) based applications like scholarly article search, recommender systems etc. Incorporating techniques for automatic topic extraction and categorization can be a major help in streamlining temporal document management in such applications.

Topic modeling is one such generative process, where techniques like association rule mining and pattern analysis can be used to discover topics (latent factor) from a large set of documents. Scholarly publication search engines are major facilitators for researchers, towards finding relevant literature, identifying research gaps etc. An important functionality that needs to be supported by such IR applications is aiding researchers in identifying relevant/related past and recent research published by peers, for discerning important focus areas for prediction of future research trends. In e-commerce, by examining the patterns of transactions for products sale and by analyzing the review comments for it, companies can anticipate and plan for future requirement of products as per demand. By examining patents submitted every year, one can identify emerging trends in product design [1]. Topic discovery can also play an important role in forecasting the behavior of stock market by analyzing the economical news [2].

Some of the critical challenges to be addressed by real-time applications are - how to summarize the large corpus and derive topics from it and how to make predictions by forecasting future trends. Past research focusing on techniques for topic discovery have mainly used clustering methods, association mining, latent semantic models and probabilistic models. The major disadvantage of these techniques is, that they consider a document as an instance of a topic, however, a document may have multiple topics, when the context/meaning of each sentence or paragraph is studied. To address this crucial research gap, we propose a novel methodology that considers sentences in a document as an instance, on which sentence-level association rule mining is applied, to capture multiple topics inherent to a single document. We considered every sentence as a transaction and each keyword or word present in sentence as an item set. We also considered frequently co-occurring patterns and focused on maximizing the context information during topic discovery. Since sentence-level topic mining is performed, more diverse and realistic topic discovery results can be achieved.

Another property of real-time data is that, current topics may not have the same importance in the future. This is especially true in research areas like computer science, bio-technology etc. To accurately measure the importance of topics over a period of time, temporal correlation analysis of topics can be very helpful. Sometimes, changes in the popularity of a topic may not be visible over several weeks or months, but over the period of years. Hence, a good forecasting approach based on multi-variant time-series analysis while predicting future topic trends is needed. This is addressed in our work, where, a sentence-level association rule mining technique based on Recursive Elimination, for discovering strongly correlated keywords as a pattern is proposed. Correlation analysis and time series analysis are also performed on discovered topics to identify emerging and future trends.

The rest of this paper is organized as follows - In Sect. 2, we present a brief discussion on related work; Sect. 3 describes the proposed technique for topic modeling and future trend forecasting, specifically applied to computer science scholarly literature. In Sect. 4, we discuss the implementation specifics, experimental setup and observed experimental results, followed by conclusion and future work in Sect. 5.

2 Related Work

Today's search engines provide results using either a predefined set of concepts or by matching documents to the keywords present in human-defined search queries. Concept-based retrieval relies on unsupervised techniques such as clustering and matching based on the user's query. Here, information can be extracted by classifying results based on predefined labels. For instance, Wordnet [3] and semantic networks [4] use predefined topics for classifying and organizing documents. IR applications also rely on keyword matching and vector-based representation based on word occurrence in documents [5].

For a given set of documents, clustering techniques use various similarity measures to separate documents in different clusters. Ayad et al. [6] employed a novel technique based on an aggregation of results obtained from various clustering algorithms, for topic discovery. From the clusters formed, they designed a technique to select the most relevant topic term from a cluster's feature space, for trending topic discovery. Yang et al. [7] designed an multi-ant colony clustering algorithm for identifying trending topics from documents. Their approach was composed of three parts, where documents are first represented in vector space, then a hypergraph is constructed to which Ant colony algorithm is applied, to recursively identify popular topics.

Pons-Porrata et al. [8] presented an incremental hierarchical clustering algorithm for news streams, that aims to summarize the news feeds and present a list of interesting topics as part of the generated hierarchies. Jayabharathy et al. [9] used labels derived from document features represented in clusters instead of using a predefined set of labels. However, the major limitation of clustering based techniques comes into light when the volume of data is high and diverse. Then, most traditional document clustering techniques fail to capture the high level semantics of the document corpora. Although different clustering techniques have been used to find out the characteristic based clusters, but the problem of identifying representative keyword has still remained a tough task.

Latent Semantic Analysis (LSA) [10] is a technique used in finding out topical similarity between two documents, although these documents do not contain a similar set of keywords. In general, the functionality used in LSA in creating a low dimensional feature space that does not directly correspond to single term, but represents it as a combination of terms in the original space [11]. In this way, LSA can help in assessing semantic similarity between documents and sorting of multi-dimensional words & manual judgments into categories. Newman et al. [12] used Probabilistic Latent Semantics Approaches (PLSA) for finding popular topics from American newspapers published in the 18th century. They used PLSA technique for the analysis of co-occurrence of topics in the document corpus. Latent Dirichlet Allocation (LDA) [13] is a technique for modeling text documents as a mixture of few semantic topics which then split words with some probability. LDA was applied for topic discovery by Zhu et al. [14], who suggested that term weight calculations are to be considered as a deterministic criterion. LDA is very similar to the probabilistic latent semantics approaches, as it was basically developed to improve the combined models and belongs to the family of Bayesian non-parametric approaches.

Time Series Analysis and Forecasting is a challenging problem and it is often difficult to obtain reasonably good forecasting results on time series based data sets. Adhikari et al. [15] and Granger et al. [16] showed experimentally that using combined forecasters with multiple methodologically different models can produce better results in topic prediction. Ensemble forecasting techniques predict topics by using ranking based approach on the basis of their in-sample absolute errors by successively ranking terms on every time series based on the past forecasting accuracy and then selectively predict results from predefined number of highly ranked models. Zhi et al. [17] presented three different ensemble forecasting models like linear regression based model, non-linear neural network based super ensemble forecasting and bias-removed ensemble mean and multi-model ensemble mean for topic prediction.

Based on the review, we observed that earlier approaches for topic modeling typically considered an entire document as a single instance due to which often fail in discovering latent topics observable at smaller granularity, e.g. paragraph or sentence level. Traditional unsupervised clustering techniques when used for topic modeling and categorization are limited to selecting only one most significant topic, thus ignoring many other important topics that may be present in the same cluster. Also, the associated computational cost grows exponentially as the volume of corpus increases. In the proposed work, we considered every sentence as a transaction, with an aim to identify all hidden latent factors. To overcome these limitations, we proposed a memory-efficient Recursive Elimination (RE) Frequent item-set based association rule mining (ARM) technique for topic discovery, specifically for computer science scholarly publications. The proposed RE algorithm is highly memory-efficient even when large document corpora are considered. Temporal correlation analysis is performed on the discovered topics for assessing the relative importance of latent topics. Time series analysis is also applied to forecast emerging trends using various regression techniques.

3 Proposed System

Figure 1 depicts the major processes followed in the proposed methodology for topic discovery and future trend forecasting in scholarly research, each of which are discussed in detail below.

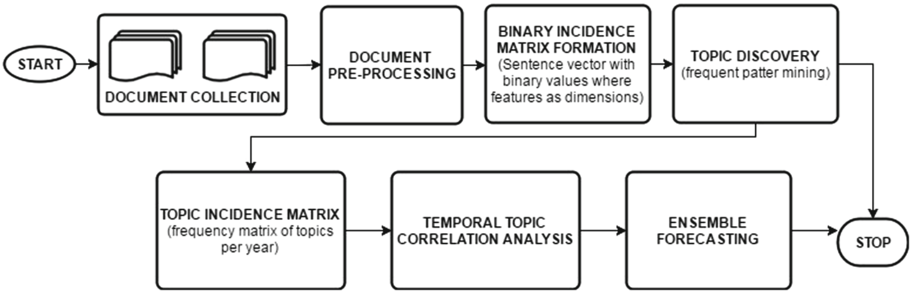

START
DOCUMENT COLLECTION
DOCUMENT PRE-PROCESSING
BINARY INCIDENCE MATRIX FORMATION
(Sentence vector with binary values where features as dimensions)
TOPIC DISCOVERY
(frequent patter mining)
TOPIC INCIDENCE MATRIX
(frequency matrix of topics per year)
TEMPORAL TOPIC CORRELATION ANALYSIS
ENSEMBLE FORECASTING
STOP

Fig. 1. Proposed methodology

3.1 Document Collection

The focus of our work is temporal topic modeling and analysis of computer science scholarly literature. Scholarly articles comprising of standard IEEE publication are collected from IEEE gateway [18]. Different queries covering different parameters and filters as per IEEE gateway specification are used, and scholarly articles in the field of computer science over the past 16 years are collected. The final corpus consists of XML documents containing information about research papers, such as - Title, Author, Affiliations, Year, Abstract etc. After extracting year-wise XML documents from IEEE Gateway, each XML document is parsed to prepare text documents representing the associated research paper. This text document contains the Title, Abstract, Author, Affiliation details of the article. The final corpus contained about 5100 documents published during the years 2000 to 2016, at the rate of about 300 documents for each year.

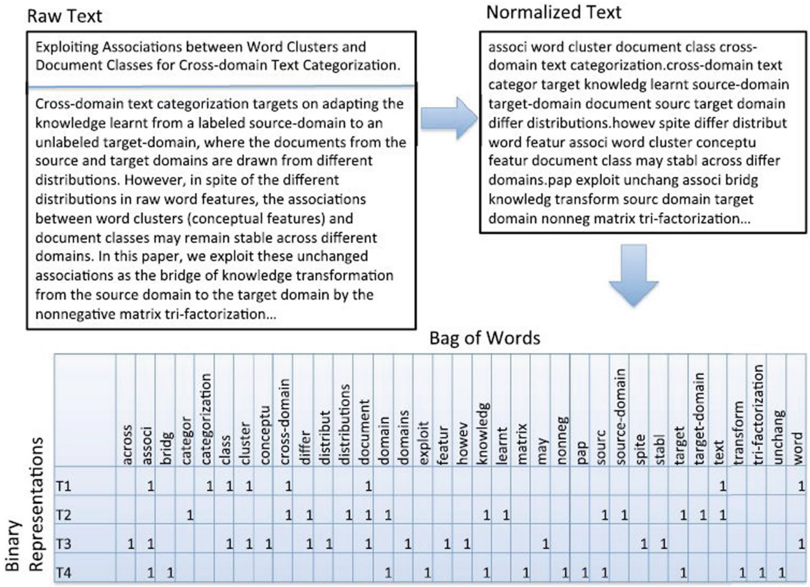
3.2 Preprocessing

To remove unwanted and noisy data from the document corpus, natural language processing (NLP) techniques [19] are used. In the initial feature space, low-value words often occur (for example, *is, am, we, thus, where, a, the, who, be, also, on* etc.), which contribute very little towards topic modeling. These stop-words are removed by using a standard English language stop-word list, thus reducing the computational complexity. As sentence level association mining is proposed, all sentences are tokenized to form a bag of words. Tokenization is a process of splitting any document into words or symbols, and removal of special characters. A token is the instance of a stream of characters that grouped together as a meaningful preprocessed unit. After tokenization, several terms are obtained, to which a Parts-of-Speech (POS) tagger [20] is applied for identifying the part of speech associated with each term. Since verbs can define only actions, not any kind of topic, we pruned all terms tagged as verbs from the sentence-level bag of words. Next, a process called stemming is used to identify the root word from the derivationally related formatted term. For example, words like *'nationalist', 'nationalism', 'national'* etc., are derived from the original root word 'nation'.

Removing these multiple terms with the same stem further reduces the original term space. The Porter Stemmer [21] was used for performing stemming on the terms obtained from the element name-phrases. Finally, the output of the preprocessing phase is a representation of each document as a *d*-dimensional sentence-level transaction vector, where each tokenized term is considered as a dimension.

3.3 Binary Vector Representation

The main purpose of the preprocessing phase is to extract all keywords by transforming each sentence into the transaction and creating its representative vector. Every keyword present in sentence vector is marked as 1 and those which are not present are marked as 0, hence these transaction vectors are represented as a Binary Matrix. Figure 2 depicts the process of transformation of a document into transactions and then into binary matrix. The abstract and title is first normalized using various NLP techniques at sentence level and then transformed into a vector space representation, where, keywords present in sentences are considered as dimensions. The binary incidence matrix is formed for each document and is then used in the process of automatically discovering similar patterns i.e., topics, from these representations.



Exploiting Associations between Word Clusters and Document Classes for Cross-domain Text Categorization.

Cross-domain text categorization targets on adapting the knowledge learnt from a labeled source-domain to an unlabeled target-domain, where the documents from the source and target domains are drawn from different distributions. However, in spite of the different distributions in raw word features, the associations between word clusters (conceptual features) and document classes may remain stable across different domains. In this paper, we exploit these unchanged associations as the bridge of knowledge transformation from the source domain to the target domain by the nonnegative matrix tri-factorization...

associ word cluster document class cross-domain text categorization.cross-domain text categor target knowledg learnt source-domain target-domain document sourc target domain differ distributions.howev spite differ distribut word featur associ word cluster conceptu featur document class may stabl across differ domains.pap exploit unchang associ bridg knowledg transform sourc domain target domain nonneg matrix tri-factorization...

	across	associ	bridg	categor	categorization	class	cluster	conceptu	cross-domain	differ	distribut	distributions	document	domain	domains	exploit	featur	howev	knowledg	learnt	matrix	may	nonneg	pap	sourc	source-domain	spite	stabl	target	target-domain	text	transform	tri-factorization	unchang	word
T1		1			1	1	1		1				1																		1				1
T2				1					1	1		1	1	1					1	1					1	1			1	1	1				
T3	1	1				1	1	1		1	1		1		1		1	1				1					1	1							1
T4		1	1											1		1			1		1		1	1	1				1			1	1	1	

Fig. 2. Vector space representation

3.4 Topic Discovery

Our topic modeling methodology aims to use every sentence in a document as a transaction. After preprocessing, each such sentence is represented in binary vector. Since transaction vectors formed are sparse in nature and may lead to higher computational cost and poor quality results hence we employed a memory efficient Frequent Itemset Mining algorithm called Recursive Elimination [22] for discovering association rules from the set of transaction vectors, that can provide an insight into popular topics.

Association Rule Mining (ARM) [23] is a well known data mining technique used to derive useful information for decision makers from raw data. Association rules can always be visualized as "implication" of the form $A \Rightarrow B$. Frequent itemset techniques use support count value and confidence value to measure the intensity or relevance of a discovered rule. Commonly used ARM terms are defined as below:

1. *Itemset:* a collection of one or more items under consideration. In our work, keywords derived from each sentence are considered as items.
2. *Support count:* the number of times an itemset is present in a transaction.
3. *Support:* defined as the fraction of the transaction that contains an itemset (given by Eq. (1)).

$$Supp(A \Rightarrow B) = \frac{|\{t \in T | A \cup B \subset t\}|}{|T|} \tag{1}$$

4. *Confidence:* the part of transaction that contains the given consequent along with the antecedent i.e. the conditional probability of B given A [24] (given by Eq. (2)).

$$Conf(A \Rightarrow B) = \frac{|\{t \in T | A \cup B \subset t\}|}{|\{t \in T | A \subset t|\}} \tag{2}$$

Frequent Itemset Generation: RElim (Frequent Itemset Mining with Recursive Elimination) algorithm is the best suited for sparse data due to its simple working principle and low memory requirement. Here, the itemsets are checked during depth-first traversal and candidate set are generated at once (similar to apriori) while maintaining one transaction list per item. Hence, it consumes less space for processing. Also, RElim is faster than Split-and-Merge technique as its time complexity is less than O(nlogn) for complete processing [25].

Figure 3 depicts the process of initial database preparation for RElim algorithm. The database transactions to be considered (Step 1), the frequency of individual items present in transaction database (step 2) and the frequent item sets sorted as per their frequency in the transaction (Step 3) as shown in Fig. 3. Step 4 shows the transactions sorted lexicographically in descending order of their item frequency.

Figure 4 shows the detailed processing performed as per the RElim algorithm: in step 1, we process the least frequent item set first (i.e. item *e* from Fig. 3), if the counter present in list is greater than the minimum support count then this item sets is treated as frequent. Step 2 shows the original list that is traversed to construct the new list array, using the leading item. In step 3, new list items are also added in the original conditional database, where, the leading item indicates the place where the succeeding items are to be added. Then, this conditional transaction database is processed recursively for deriving all other frequent itemsets.

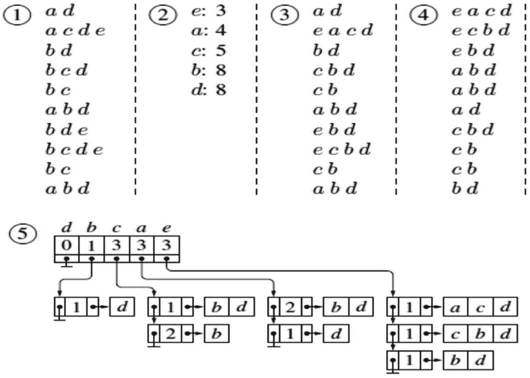


Fig. 3. A sample transaction database preprocessed using the RElim algorithm

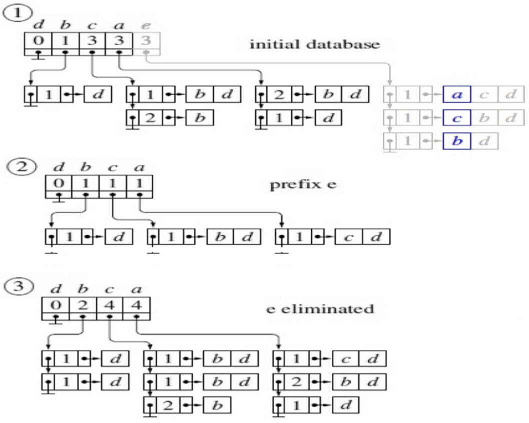


Fig. 4. Basic operations of RElim algorithm

Rule Generation and Refinement: From intermediate results, we observed that discovered topics with a length greater than three, i.e., four-grams, five-grams etc. are very much irrelevant w.r.t desired topics (for instance, *wireless_network_sensor_wsn, svm_support_vector_machin, signal_snr_nois_ratio, frequenc_multiplex_ofdm_divis, divis_frequenc_multiplex_ofdm_orthogon* etc.). Hence, only bi-gram and tri-gram topics discovered are retained from the initially discovered set of topics. We have applied NLP techniques and extracted list of bi-grams and tri-grams from original documents and matched them with discovered topics, the match which has produced difference zero is considered as a meaningful topic phrase.

Rules generated or the topics discovered using association rule mining may contain repeated patterns; hence all discovered rules cannot be directly considered as a possible topic. To prune such redundancies, set operations are used to refine such association rules into a single rule. As per set theoretic concepts, if the discovered rule is a proper subset of another rule, then all such proper subset rules which are discovered as a topic are removed from consideration. For instance, if two phrases *ad_hoc* and *ad_hoc_network* were found, then *ad_hoc* is filtered out as it is the proper subset of the other discovered rule. These techniques help in refining discovered association rules and were found to produce better results.

3.5 Topic Incidence Matrix Generation

For the discovered set of topics, the topic incidence matrix is generated using the dataset containing documents collected for last 17 years i.e. from year 2000 to 2016. The frequency of each discovered topic is computed and used to generate the 2D vector which is further used in temporal correlation analysis and ensemble forecasting. Table 1 shows a sample output obtained for some discovered topics.

Table 1. Topic incidence matrix

Topics vs. years	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
support_vector_machin	1	5	6	4	4	5	4	1	3	5	3
ad_hoc_network	3	4	7	2	5	5	8	2	6	3	12
orthogon_frequenc_divis	2	0	0	0	0	4	4	1	3	3	5
particl_swarm_optim	3	1	3	3	3	2	5	3	0	4	0
hidden_markov_model	0	0	2	2	1	2	0	2	4	1	0
ant_coloni_optim	1	1	2	1	2	0	1	4	0	0	0
artifici_neural_network	6	2	1	1	1	0	1	3	0	0	1

3.6 Temporal Topic Correlation Analysis

The topic vectors obtained after applying the topic discovery process are used for finding the correlation between all topics. Pearson's Correlation Coefficient (PCC) was used to find correlation between two topic vectors. Consider two random vectors X and Y with all their observed values $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$. Then, correlation between X and Y can be calculated using Eq. (3).

$$\rho(X, Y) = \frac{Cov(X, Y)}{\sqrt{Var(X)Var(Y)}} \tag{3}$$

where, $Cov(X, Y)$ is the covariance between X and Y and $Var(X)$ and $Var(Y)$ is the variance. *Covariance* is the measure of how two random variables X and Y

change together and is given by Eq. (4). *Variance* refers to the spread of dataset i.e. how far apart the numbers are in relation to the mean, for instance. The sample variance of random variable X is defined as per Eq. (5).

$$Cov(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1} \tag{4}$$

$$Var(X) = \frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n-1} \tag{5}$$

Therefore, correlation analysis between two random variables can be performed using PCC as per the Eq. (6).

$$\rho(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 * \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{6}$$

The value of the Pearson's correlation measure lies within the range of +1 to −1. For any two random variables, a negative correlation value indicates that there is an inverse relationship between two variables, i.e., if one variable increases other variable will decrease and vice versa. A positive correlation value indicates a direct relationship between two variables, while a value of 0 indicates that two variables are independent of each other. Table 2 tabulates some correlation values obtained for a few sample discovered topics.

Table 2. Correlation analysis for some discovered topics

Topics	support-vector-machin	ad-hoc-network	orthogon-frequenc-divis	particl-swarm-optim	hidden-markov-model	ant-coloni-optim	artifici-neural-network
support-vector-machin	1	0.15	−0.02	0.42	−0.27	−0.14	−0.22
ad-hoc-network	0.15	1	0.33	−0.06	0.12	−0.24	−0.41
orthogon-frequenc-divis	−0.02	0.33	1	0.11	−0.19	−0.28	−0.37
particl-swarm-optim	0.42	−0.06	0.11	1	−0.31	0.44	−0.25
hidden-markov-model	−0.27	0.12	−0.19	−0.31	1	−0.01	−0.32
ant-coloni-optim	−0.14	−0.24	−0.28	0.44	−0.01	1	−0.02
artifici-neural-network	−0.22	−0.41	−0.37	−0.25	−0.32	−0.02	1

3.7 Topic Trend Forecasting

The proposed forecasting algorithm takes the target topic and applies time series analysis to its historical data to forecast its future trends. Our approach provides both single-level and multi-level prediction, i.e., multiple topics can be forecasted by considering the relative historical time series data of more than one topic at a time. The advantage of multi-topic techniques over single topic prediction is that, a temporal correlation within different topics can help to improve the accuracy of the forecast. Since topics selected for forecasting can significantly affect results, it is very important to know which supporting topics should be considered in order to forecast targeted topic's future value [26]. In our work, we used ensemble forecasting techniques where multiple forecasters are combined for forecasting future trends in varied topics. We used the WEKA forecast tool [27] along with a set of regression learning methods to model and predict time series data. Three different forecasting models are used - Linear Regression Model (LR), Support Vector Regression (SVR) and a weighted ensemble of LR and SVR. These models are discussed in detail below.

1. Linear Regression Model (LR): Linear regression is an associative model, that aims to predict the score of a dependent variable using an independent variable. The dependent variable to be predicted is called the *criterion variable* and is generally plotted on Y-axis, where as the independent variable is known as the predictor variable and is generally plotted on X-axis. Linear regression is the process of finding the best-fit straight line though the points on $X - Y$ plane such that the vertical distance of the point from the regression line (error) is the least. Figure 5 shows a LR scatter plot for sample data, where the best fitting line for the points (called regression line) is shown. The LR model helps in determining the slope of regression line, which is used in finding the value of the criterion variable, i.e. the target topic. The slope of the regression line is calculated as per Eq. (7).

$$Y = a + bX \tag{7}$$

where, Y is the dependent variable (predicted future trending topic) on independent variable X (target topic provided by the user), a is the intersection of regression line at y-axis and b is the slope of line or some constant.

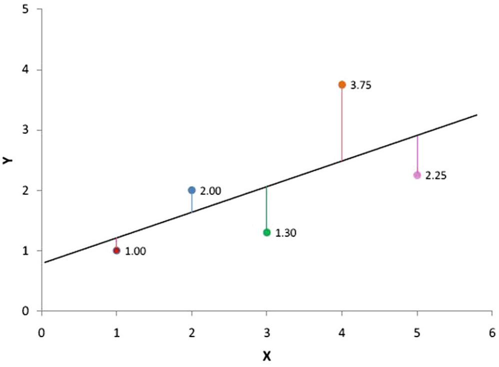


Fig. 5. Scatter plot of sample data

2. Support Vector Regression Model (SVR): SVM for regression was proposed in 1997 by Vapnik et al. [28]. In SVR, a function $f(x)$ that has at most deviation ϵ from the actually obtained targets y_i for all the training data is to be found. SVR attempts to minimize the generalization error bound so as to achieve generalized performance, rather than minimizing the observed training error. In regression problems, we are given a set of training data

$D = (x_i, y_i) \,|\, x_i \epsilon R^n, \;\; y_i \epsilon R, \;\; i = 1, ..., l$. Suppose that the data is collected from the model as in Eq. 8:

$$Y_i = f(x_i) + \delta_i \tag{8}$$

where, $f(x)$ is the underlying function and δ_i are independent and identically distributed random noises. Here $f(x)$ can be defined as $f(x) = <\omega, x> + b$, where $< \ldots, \ldots >$ denote the dot product in R^n. Flatness indicates a small value of ω. In WEKA, the kernel type "PolyKernel" was used to fit the data using curved line and set C i.e. complexity value was set as 1.0. This allows some violations while drawing the line to fit data, for different values of parameter forecasting value and error value changes.

3. Weighted ensemble of LR & SVR (Ensemble): An equal weighted ensemble of LR and SVR is taken as the third model. Figure 6 illustrates the process of ensemble forecasting. For obtaining a forecast topic X_i from all forecasters, N randomly chosen topics excluding X_i are considered initially. To overcome biasness of classifiers, we considered M forecasters and apply an aggregation to their forecast values using a weighted averaging method. Hence, finally forecast results are the average of all forecasters' output. The primary objective is to check how many randomly chosen topics (N) can yield the best forecast value for the given target topic.

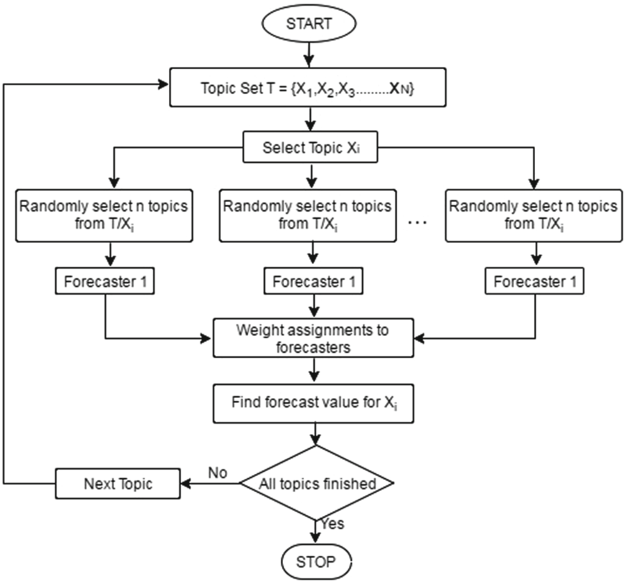

START
Topic Set T = {X1,X2,X3.........XN}
Select Topic Xi
Randomly select n topics from T/Xi
Randomly select n topics from T/Xi
...
Randomly select n topics from T/Xi
Forecaster 1
Forecaster 1
Forecaster 1
Weight assignments to forecasters
Find forecast value for Xi
All topics finished
No
Next Topic
Yes
STOP

Fig. 6. Proposed ensemble forecasting framework

4 Experimental Evaluation and Results

To validate the proposed methodology, experiments are carried out on a dataset of computer science scholarly publications over the period of 17 years, collected from IEEE gateway. These were preprocessed as discussed earlier in Sect. 3 and the proposed topic discovery mechanisms were applied.

4.1 Topic Discovery

For topic discovery, the Recursive Elimination based frequent itemset mining algorithm is applied and association rules with different support and confidence values are derived. Bigram and trigram models are used for representing the discovered topics, as these are found to be most suitable for discovered computer science topics. These bigrams and trigrams are used to filter out redundant association rules by performing mathematical set theory operations. We also removed proper subsets and considered only supersets which produced refined association rules. Table 3 shows the number of discovered topics for different support and confidence value. We used different document counts and for different confidence and support values to evaluate the goodness of the model. It is observed that most relevant topics are discovered using support = 20 and confidence = 0.8 when input document size was 3400. Some such discovered topics after this process are shown in Table 4. It can be observed from Table 4, that topics discovered are relevant and meaningful.

Table 3. Experimental statistics with different support and confidence value

Document-count	Support	Confidence	Discovered-topic-count
3400 articles	13	0.85	129
	13	0.90	98
	13	0.95	53
	14	0.95	46
	15	0.9	75
	20	0.8	69
5100 articles	13	0.9	187
	15	0.9	132
	20	0.9	81

Consider a target word *'ad_hoc_network'*. From the set of discovered topics, the topics such as *neural_network_time*, *mobil_ad_hoc* and *support_vector_classifi* with correlation values (0.61, 0.61, 0.62) respectively, were found to be strongly correlated to *'ad_hoc_network'*. We also considered strongly negatively correlated topics viz. *radial_basi_function*, *neural_network_learn* and *artifici_neural_network* with correlation values (−0.61, −0.38, −0.41) respectively. In Table 5, the actual

Table 4. Discovered topics

Sample topics discovered		
local_wireless_network	hoc_network_mobil	bit_error_rate
function_neural_network	compar_state_art	signal_ratio_snr
support_vector_machin	svm_support_vector	mobil_ad_hoc
kalman_filter_estim	sensor_network_wsn	protocol_ad_hoc
ad_hoc_network	slide_mode_control	divis_multipl_access
close_loop_control	control_neural_network	algorithm_neural_network
genet_algorithm_optim	signal_nois_ratio	area_wireless_network
orthogon_frequenc_divis	neural_network_paper	spectrum_cognit_radio
space_time_code	perform_state_art	perform_neural_network
intrus_detect_system	radial_basi_function	hoc_wireless_network
genet_algorithm_ga	frequenc_divis_multiplex	ldquo_rdquo
particl_swarm_optim	method_neural_network	sensorless_control
input_multipl_output	neural_network_model	worst_case
neural_network_approach	neural_network_learn	spatio_tempor
neural_network_featur	network_mobil_ad	duti_cycl
network_wireless_sensor	paper_genet_algorithm	simul_anneal
neural_network_data	classif_neural_network	steadi_state
neural_network_time	wireless_ad_hoc	problem_np
hidden_markov_model	fuzzi_neural_network	cosin_transform
time_domain_finit	optim_problem_solv	mont_carlo
qualiti_servic_qo	neural_network_train	princip_compon
hoc_network_protocol	artifici_neural_network	pid_control

frequencies of discovered topics over the period of 17 years i.e. 2000–2017 are tabulated. These values are used in forecasting model to forecast topic trend 1-step ahead. Table 6 shows the forecasting analysis for topic *ad_hoc_network* performed in the six different scenarios described earlier. Figure 9 compares the forecasting performance of the three different models for the six scenarios considered.

4.2 Ensemble Forecasting

The effectiveness of each forecaster is evaluated using the metrics Mean Squared Error (MSE), Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Lower the value of each of these metrics, better the performance of the forecasting framework. *MSE* [29] is the average of squares of difference between actual values and the predicted values (Eq. 9). *RMSE* [29] is a measure of closeness of data to the regression line (Eq. 10). *MAE* [30] is used to measure how close the forecasts or predictions are to the eventual outcomes (Eq. 11).

Table 5. Year-wise actual and forecasted frequency values of Discovered Topics (here T1–T7 are topics; T1 = 'ad_hoc_network' and Year 2000–2016 are represented as 00–16)

Topic v/s Yr	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
T1	0	3	3	7	14	7	3	4	7	2	5	5	8	2	6	3	12
T2	0	0	0	0	2	1	0	0	0	0	0	0	2	0	0	0	0
T3	0	2	2	1	10	2	0	0	9	0	4	1	1	1	7	0	2
T4	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0
T5	6	1	1	0	0	0	4	0	2	3	1	0	0	3	0	0	0
T6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T7	4	4	8	3	1	2	6	2	1	1	1	0	1	3	0	0	1

$$MSE = \frac{\sum_{i=1}^{n}(Y_i - \bar{Y}_i)^2}{n} \tag{9}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Y_i - \bar{Y}_i)^2}{n}} \tag{10}$$

where, Y is actual value, $\overline{Y}$ is predicted value & n is total predictions made.

$$MAE = \frac{\sum_{i=1}^{n}|f_i - \bar{y}_i|}{n} \tag{11}$$

where, f is prediction and y is true value.

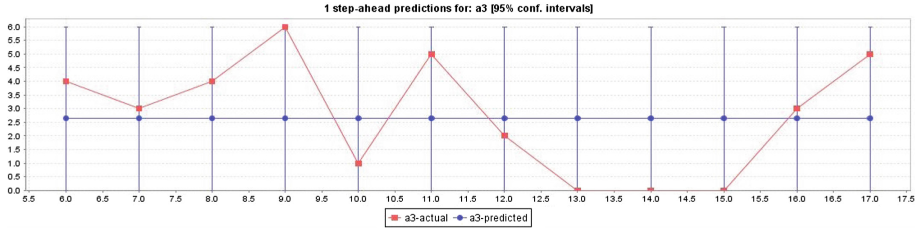


Fig. 7. Forecasting with Linear Regression (LR) model

Figures 7 and 8 depict a plot of actual values and predicted values generated by the three models - LR, SVR and LR+SVR regression models, applied on the set of discovered topics for time series analysis. It can be clearly observed that LR prediction contained more erroneous values when compared to SVR. Values predicted by SVR are more closer to the actual values. For comprehensive evaluation of the forecasting performance, different scenarios were defined, as listed below. For each of these scenarios, the forecasting performance of these three models was observed and compared.

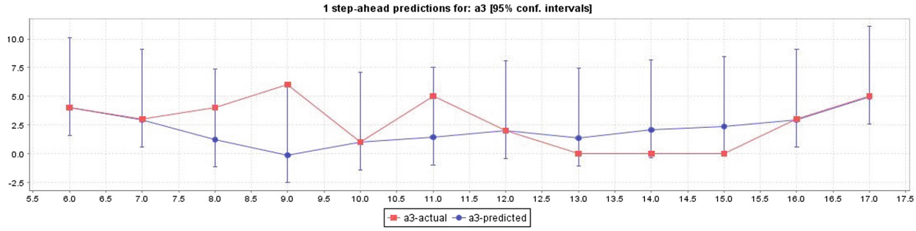


Fig. 8. Forecasting with Support Vector Regression (SVR) model

1. *Single topic:* only one target topic is considered for future trend forecasting.
2. *Highest correlated:* the topic with highest Pearson coefficient value is considered along with target topic.
3. *Highly correlated:* In this case, two or more highly correlated topics are considered together. For evaluation purposes, the threshold value for high correlation was fixed as 0.5.
4. *Highest negatively correlated:* The topic with least Pearson coefficient value is considered in this scenario along with the target topic.
5. *Inversely correlated:* all topics with Pearson's correlation coefficient value less than −0.3 are considered.
6. *Random topics:* A fixed number of random topics are chosen for forecasting.

Figures 10 and 11 depict the plots of the error values generated by different models. From these plots, it can be concluded that LR model produced more error than SVR. Figure 11 clearly indicates that the scenario *highly correlated* produced better forecast values when compared to the other scenarios; where as

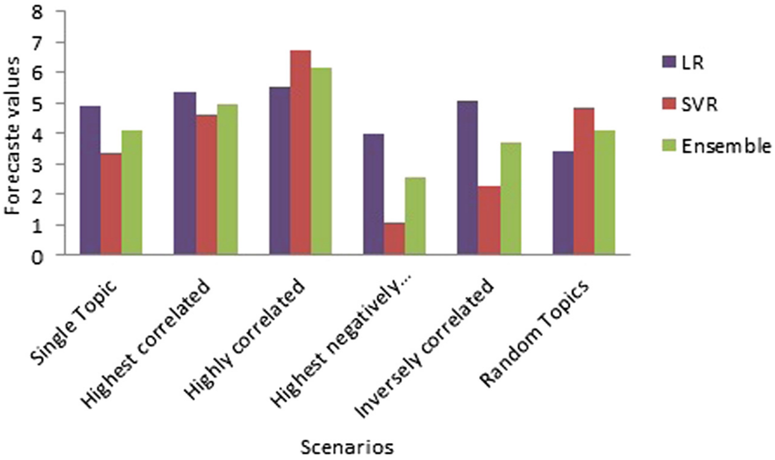


Fig. 9. Forecast values for LR, SVR and ensemble models for the six scenarios considered for evaluation

Table 6. Forecasting analysis for topic ad_hoc_network

Scenario	Model	MSE	RMSE	MAE	Forecasted value
1: Single topic	LR	6.4879	2.5471	2.0535	4.9036
	SVR	2.0392	1.428	0.9220	3.3607
	Ensemble	4.2635	1.9875	1.4877	4.1321
2: Highest correlated	LR	7.7226	2.779	2.2255	5.3529
	SVR	1.5862	1.2594	0.6705	4.6319
	Ensemble	4.6500	2.0200	1.4500	4.9900
3: Highly correlated	LR	7.6000	2.758	2.1859	5.5123
	SVR	0.0020	0.0155	0.0142	6.7534
	Ensemble	3.8010	1.3867	1.1000	6.1328
4: Highest negatively correlated	LR	3.9192	1.9797	1.4885	4.0035
	SVR	2.165	1.4714	0.9353	1.0849
	Ensemble	3.0400	1.7300	1.2100	2.5400
5: Inversely correlated	LR	7.7915	2.7913	2.1784	5.0702
	SVR	0.6557	0.8098	0.2980	2.3049
	Ensemble	1.2382	1.8005	1.2384	3.6875
6: Random topics	LR	5.0466	2.2465	1.5228	3.3939
	SVR	0.2440	0.4940	0.1559	4.8412
	Ensemble	2.6453	0.8762	0.8393	4.1175

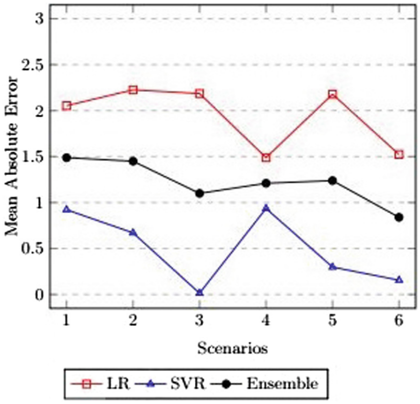


Fig. 10. Comparison of forecasting models based on MSE

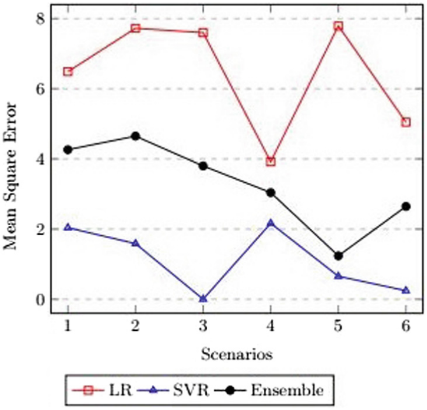


Fig. 11. Comparison of forecasting models based on MAE

the scenario *highest inversely correlated* fared the worst. This supports the belief that strongly positively correlated topics that support each other are likely to be used more together. Due to the variations in the considered forecasting models, the ensemble forecasting technique can be considered to be the best forecasting technique to overcome the drawbacks of different forecasting models. Also, using a weighted ensemble helps in normalizing the forecast values to some scale.

5 Conclusion and Future Work

In this work, a topic modeling and discovery approach based on the Recursive Elimination based association rule mining approach for topic trend forecasting for computer science scholarly publications is presented. Experiments are carried out on IEEE publications in the area of computer science, over a period of 17 years. To understand the document context at sentence level, each article is processed to extract sentences and these sentences are represented as transactions for mining frequent itemsets and association rules. Further, set inclusion/exclusion operations and a dictionary of bi-grams and tri-grams from input documents are used for identifying correct topics and pruning redundant ones. Temporal correlation analysis is performed on the discovered topics using two different regression models, based on which ensemble forecasting is performed on a target topic. It was observed that the Support Vector Regression model performed better than the Linear Regression model, for most considered scenarios. Further, temporal correlation analysis can be further enhanced by supporting the analysis of latent relationships between interdisciplinary fields, thus helping in promoting interdisciplinary research. We also intend to build an academic search engine with functionalities based on the discussed methodology, on the premise of a personalized recommender system intended for researchers and authors.

References

1. Tucker, C., Kim, H.: Predicting emerging product design trend by mining publicly available customer review data, vol. 6, pp. 43–52 (2011)
2. Schumaker, R.P., Chen, H.: Textual analysis of stock market prediction using breaking financial news: the AZFin text system. ACM Trans. Inf. Syst. **27**(2), 12:1–12:19 (2009). http://doi.acm.org/10.1145/1462198.1462204
3. Liu, Y., Scheuermann, P., Li, X., Zhu, X.: Using WordNet to disambiguate word senses for text classification. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2007. LNCS, vol. 4489, pp. 781–789. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72588-6_127
4. Sussna, M.: Word sense disambiguation for free-text indexing using a massive semantic network. In: Proceedings of the Second International Conference on Information and Knowledge Management, CIKM 1993, pp. 67–74. ACM, New York (1993). http://doi.acm.org/10.1145/170088.170106
5. Wiemer-Hastings, P., Wiemer-Hastings, K., Graesser, A.: Latent semantic analysis. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence, pp. 1–14. Citeseer (2004)

6. Ayad, H., Kamel, M.: Topic discovery from text using aggregation of different clustering methods. In: Cohen, R., Spencer, B. (eds.) AI 2002. LNCS (LNAI), vol. 2338, pp. 161–175. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-47922-8_14
7. Yang, Y., Kamel, M., Jin, F.: Topic discovery from document using ant-based clustering combination. In: Zhang, Y., Tanaka, K., Yu, J.X., Wang, S., Li, M. (eds.) APWeb 2005. LNCS, vol. 3399, pp. 100–108. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31849-1_11
8. Pons-Porrata, A., Berlanga-Llavori, R., Ruiz-Shulcloper, J.: Topic discovery based on text mining techniques. Inf. Process. Manag. **43**(3), 752768 (2007)
9. Jayabharathy, J., Kanmani, S., Parveen, A.A.: Document clustering and topic discovery based on semantic similarity in scientific literature. In: 2011 IEEE 3rd International Conference on Communication software and networks (ICCSN), pp. 425–429. IEEE (2011)
10. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**(6), 391 (1990)
11. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. Discourse Process. **25**(2–3), 259–284 (1998)
12. Newman, D.J.: Probabilistic topic decomposition of an eighteenth-century American newspaper. J. Am. Soc. Inf. Sci. Technol **57**, 753–767 (2006)
13. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
14. Zhu, D., Fukazawa, Y., Karapetsas, E., Ota, J.: Intuitive topic discovery by incorporating word-pairs connection into LDA. In: 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, December 2012, vol. 1, pp. 303–310 (2012)
15. Adhikari, R., Verma, G., Khandelwal, I.: A model ranking based selective ensemble approach for time series forecasting. Procedia Comput. Sci. **48**, 14–21 (2015)
16. Granger, C.W., Ramanathan, R.: Improved methods of combining forecasts. J. Forecast. **3**(2), 197–204 (1984)
17. Zhi, X., Qi, H., Bai, Y., Lin, C.: A comparison of three kinds of multimodel ensemble forecast techniques based on the tigge data. Acta Meteorologica Sinica **26**, 41–51 (2012)
18. Senn, M.: IEEE explorer gateway (2009). http://ieeexplore.ieee.org/gateway. Accessed 20 Oct 2016
19. Loper, E., Bird, S.: NLTK: the natural language toolkit. In: Proceedings of the ACL 2002 Workshop on Effective tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, vol. 1, pp. 63–70. Association for Computational Linguistics (2002)
20. Toutanova, K., Manning, C.D.: Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics, vol. 13, pp. 63–70. Association for Computational Linguistics (2000)
21. Porter, M.F.: An algorithm for suffix stripping. Program **14**(3), 130–137 (1980)
22. Borgelt, C.: Keeping things simple: finding frequent item sets by recursive elimination. In: Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations, pp. 66–70. ACM (2005)
23. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. ACM SIGMOD Rec. **22**(2), 207–216 (1993). ACM

24. Ruiz, M.D., Gomez-Romero, J., Molina-Solana, M., Campana, J.R., Martn-Bautista, M.J.: Meta-association rules for mining interesting associations in multiple datasets. Appl. Soft Comput. **49**, 212–223 (2016)
25. Borgelt, C.: Simple algorithms for frequent item set mining. Adv. Mach. Learn. **II**(263), 351–369 (2010)
26. Mendes-Moreira, J., Soares, C., Jorge, A.M., Sousa, J.F.D.: Ensemble approaches for regression: a survey. ACM Comput. Surv. (CSUR) **45**(1), 10 (2012)
27. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD Explor. Newsl. **11**(1), 1018 (2009)
28. Vapnik, V., Golowich, S.E., Smola, A., et al.: Support vector method for function approximation, regression estimation, and signal processing. In: Advances in Neural Information Processing Systems, pp. 281–287 (1997)
29. Willmott, C.J.: On the validation of models. Phys. Geogr. **2**(2), 184–194 (1981)
30. Maragos, P.: Morphological correlation and mean absolute error criteria. In: International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1989, pp. 1568–1571. IEEE (1989)

Trip Planning and Scheduling Queries in Spatial Databases: A Survey

Tanzima Hashem(✉) and Mohammed Eunus Ali

Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology,
Dhaka 1000, Bangladesh
{tanzimahashem,eunus}@cse.buet.ac.bd

Abstract. Planning and scheduling trips in an optimized manner allow users to perform their daily activities with convenience. A trip planning query finds a trip for a single user or a group jointly visiting different types of points of interests (POIs) such as a restaurant, a pharmacy and a movie theater with the minimum travel cost, whereas a trip scheduling query distributes the tasks of visiting different POI types among the group members by computing individual trips for the group members. In recent years, researchers have proposed variants of location based trip queries that include single trip planning queries, group trip planning queries, group trip scheduling queries, obstructed trip planning queries, dynamic group trip planning queries, and privacy preserving trip planning queries. Processing trip planning and scheduling queries in real time is a computational challenge as trips may involve more than one user and POIs of multiple types, and more importantly, the query answer is evaluated from a huge POI database. In this survey, we give an overview of the state of the art approaches for processing trip planning and scheduling queries. We compare these approaches from different angles like the number of users involved in a query (i.e., single or group), the type of the data space (i.e., Euclidean space/road networks/obstructed space), the sequence of POI types (i.e., fixed/flexible), static or dynamic, optimization parameters (i.e., distance/popularity) and privacy.

1 Introduction

Location based applications optimize the utilization of transport resources, reduce fuel consumption, and allow people to plan their daily activities with convenience. Trip planning and scheduling are important class of location based services (LBSs) that allow users to visit a number of points of interests (POIs) such as a restaurant, a pharmacy and a movie theater. Trip planning may involve a single user or a group. For example, a tourist traveling from a tourist attraction to a hotel may want to visit an ATM booth and a restaurant that together minimizes the user's trip distance or a group of friends located at different source locations may want to have a common trip with the minimum aggregate trip distance for visiting a restaurant, a shopping mall and a movie theater

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 164–178, 2017.
https://doi.org/10.1007/978-3-319-72413-3_11

together before travelling to their individual destinations. On the other hand, a trip scheduling query distributes the tasks of visiting different POI types among the group members by computing individual trips for the group members with an aim to minimize the aggregate trip distance. In this paper, we study the state of the art for processing trip planning and scheduling queries.

Processing trip planning and scheduling queries in real time is a computational challenge as trips may involve more than one user and POIs of multiple types, and more importantly, the query answer is evaluated from a huge POI database. Processing a trip planning query has been shown as a NP-hard problem in [9,23] if the sequence of visiting POI types (e.g., first a restaurant then a movie theater) is not fixed. The efficiency of a query processing algorithm depends on the number of POIs searched for identifying the optimal answer and the trip computation technique using the POIs in the refined search space. Thus, efficient indexing techniques to quickly access only the relevant POIs and efficient pruning techniques to refine the search space effectively have been proposed. Moreover, since most of the major variations of these queries are NP-Hard, several approximation techniques have also been proposed.

Trip planning and scheduling queries are different from the traveling salesman problem [22,31] and its variants [13,37]. The work done to address the traveling salesman problem and its variants assume a small number of POIs and do not propose any POI search space refinement technique. Thus, the solutions for the traveling salesman problem and its variants are not extendable for finding the answers for trip planning and scheduling queries in real time. A trip planning query is also different from a nearest neighbor query [32] or a group nearest neighbor query [3,29,30,38] in the spatial databases. Since the trip planning query involves visiting more than one POI types, and the nearest neighbor query and its variants can handle only one POI type, these existing methods for nearest neighbor queries are not applicable for trip queries.

The remainder of this paper is organized as follows. In Sect. 2, we discuss the variants of trip planning and scheduling queries that exist in the literature. In Sect. 3, we give an overview of the existing approaches for processing trip planning and scheduling queries. In Sect. 4, we compare the existing approaches from different angles like the number of users involved in a query (i.e., single or group), the type of the data space (i.e., Euclidean space/road networks/obstructed space), the sequence of POI types (i.e., fixed/flexible), static or dynamic, optimization parameters (i.e., distance/popularity), and privacy. Finally, in Sect. 5, we conclude the study with future directions.

2 Preliminaries

In this section, we formally define variants of trip planning and scheduling queries in spatial databases. Let P represent a set of POI sets $\{P_1, P_2, \ldots, P_x\}$, where P_i for $1 \leq i \leq x$ is a set of POIs of type i: $\{p_i^1, p_i^2, \ldots, p_i^j\}$. The POIs are indexed in the database of the service provider. The service provider evaluates the query using the stored POIs in the database.

Suppose Function $dist(.,.)$ returns the distance between two points in the Euclidean space, road networks, and the obstructed space. In the Euclidean space, the distance is the length of the straight line connecting two points. A road network is represented using a graph, where vertices represent road junctions and the edge between two vertices exists if there is a direct road connection between them. The distance between two points in a road network is the length of the shortest path connecting the points, where a path is a sequence of edges. An obstructed space represents the pedestrian scenario and assumes the presence of obstacles like private buildings, fences and rivers in the Euclidean space. The obstructed distance between two points is the length of the path by avoiding the obstacles.

The trip distance $Tdist(s_i, d_i, A)$ is the distance between s_i to d_i via the POIs in A, where s_i and d_i represent the user's source and destination locations, and A includes a single POI from each POI type visited by the user. The sequence of visiting POI types can be fixed or flexible. For a fixed sequence of visiting POI types: first the restaurant and then the movie theater, the trip distance is computed as the summation of distances from the source to the restaurant, from the restaurant to the movie theater, and the movie theater to the destination. On the other hand, if the sequence of POI types is flexible, e.g., the user can visit either the restaurant or the movie theater first then the trip distance is measured for the combination of POI types that provides the smallest trip distance.

The aggregate trip distance for a group is computed as the summation or the maximum of the trip distances of the group members. Let $GTdist(S, D, A)$ represent the group trip distance for a set of n source locations $S = \{s_1, s_2, ..., s_n\}$, a set of n corresponding destination locations $D = \{d_1, d_2, ..., d_n\}$ of the group members. For an aggregate function $f = \text{SUM}$, $GTdist(S, D, A) = \sum_{i=1}^{n} Tdist(s_i, d_i, A)$ and for an aggregate function $f = \text{MAX}$, $GTdist(S, D, A) = \max_{i=1}^{n} Tdist(s_i, d_i, A)$.

2.1 Trip Planning Queries

A trip planning query is formally defined as follows:

Definition 1 ***(Trip Planning Queries).*** *Given a set of POI sets P, a source s_i, a destination d_i, a set of m POI types $T = \{t_1, t_2, ..., t_m\}$, the trip planning query returns a POI set A such that $Tdist(s_i, d_i, A) \leq Tdist(s_i, d_i, A')$, where $A \neq A'$, and both A and A' include a POI from P for every POI type in T.*

A trip planning query can be extended to a k trip planning query that returns k POI sets with the k smallest trip distances. If the sequence of visiting POI types is fixed, then the trip planning query is also known as the optimal sequenced route planning query. If a sequenced route planning query is evaluated in the obstructed space then it is known as the optimal obstructed sequenced route planning query. For a privacy preserving trip planning query, a user's actual source and destination locations are not disclosed to a service provider.

Examples of trip planning queries in an Euclidean space, a road network space, an obstructed space are shown in Figs. 1, 2, and 3, respectively. There are

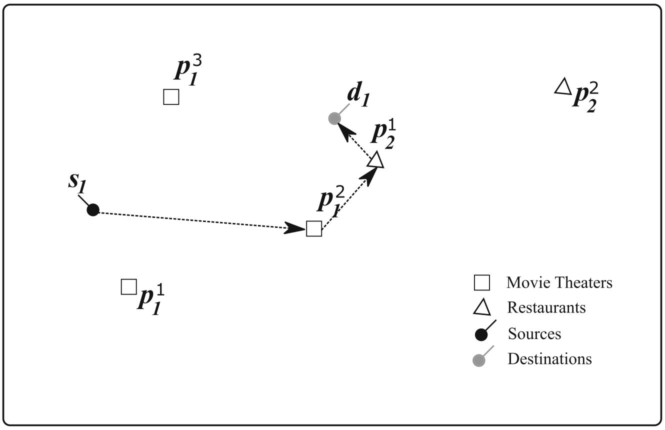


Fig. 1. A trip planning query in an Euclidean space

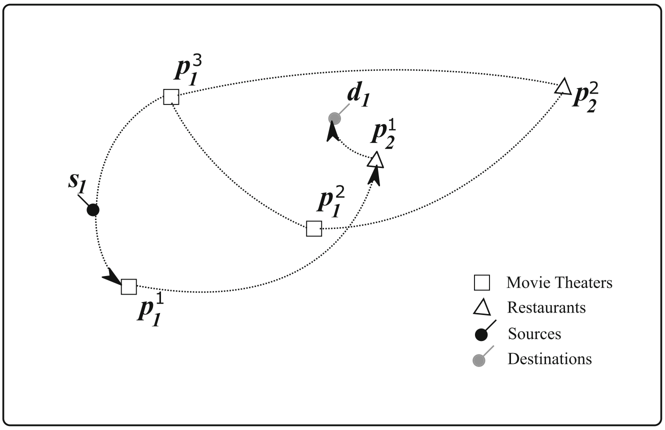


Fig. 2. A trip planning query in a road network space

5 POIs in this example, where three POIs p_1^1, p_1^2, p_1^3 are of type 1 (i.e., Movie Theatres), and two POIs p_2^1, p_2^2 are of type 2 (i.e., Restaurants). A user wants to make a trip from her source s_1 to destination d_1 passing through one POI of each type with minimum aggregate travel distance. Figure 1 shows the trip $s_1 \rightarrow p_1^2 \rightarrow p_2^1 \rightarrow d_1$ that minimizes the aggregate travel distance of the user in an Euclidean space. Similarly, Fig. 2 shows the trip $s_1 \rightarrow p_1^1 \rightarrow p_2^1 \rightarrow d_1$ that minimizes the aggregate travel distance of the user in a road network space, where the user can travel only through the roads. Finally, Fig. 3 shows the trip $s_1 \rightarrow p_1^3 \rightarrow p_2^1 \rightarrow d_1$ that minimizes the aggregate travel distance of the user in an obstructed space, where the user travels through the space by avoiding obstacles.

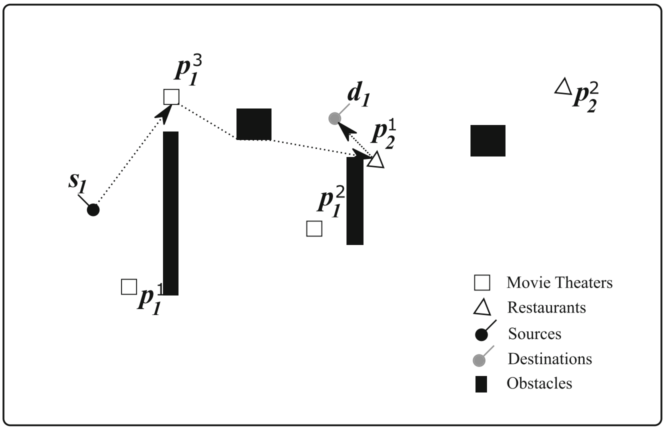


Fig. 3. A trip planning query in an obstructed space

2.2 Group Trip Planning Queries

A group trip planning query is formally defined as follows:

Definition 2 ***(Group Trip Planning Queries).*** *Given a set of POI sets P, a set of n source locations $S = \{s_1, s_2, ..., s_n\}$, a set of n corresponding destination locations $D = \{d_1, d_2, ..., d_n\}$ of the group members, a set of m POI types $T = \{t_1, t_2, ..., t_m\}$, and an aggregate function f, the group trip planning query returns a POI set A such that $GTdist(S, D, A) \leq GTdist(S, D, A')$, where $A \neq A'$, and both A and A' include a single POI from P for every POI type in T.*

Figure 4 shows an example of a group trip planning query in an Euclidean space, where there are 5 POIs: three POIs p_1^1, p_1^2, p_1^3 are of type 1 (i.e., Movie Theatres), and two POIs p_2^1, p_2^2 are of type 2 (i.e., Restaurants), and a group of three users with source locations s_1, s_2, s_3 and destination locations d_1, d_2, d_3. The group trip query returns a pair (p_1^2, p_2^1), one POI from each type, which results in the minimum aggregate trip distance for the group.

Similar to a trip planing query, a group trip planning query can be extended to a k group trip planning query that returns k POI sets with the k smallest aggregate trip distances. An important variant of the group trip planning query is the dynamic group trip planing query where members can join at any intermediate POI or leave after visiting any POI. Another important variant of a group trip planning query is a subgroup trip planning query, which is formally defined as follows:

Definition 3 ***(Subgroup Trip Planning Queries).*** *Given a set of POI sets P, a group G of n users $\{u_1, u_2, \ldots, u_n\}$, the minimum subgroup size n', a set of n corresponding destination locations $D = \{d_1, d_2, ..., d_n\}$ of the group members, a set of m POI types $T = \{t_1, t_2, ..., t_m\}$, and an aggregate function f, the subgroup trip planning query returns for every subgroup size $n'' \in [n', n]$, a subgroup $G' \in G$ of n'' users and a POI set A that together minimizes the group*

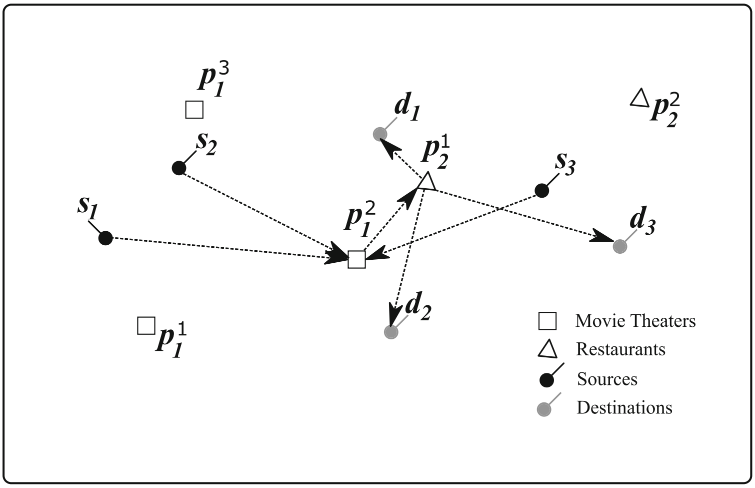


Fig. 4. A group trip planning query in an Euclidean space

trip distance for the subgroup size n'', where A includes a POI from P for every POI type in T.

For the example shown in Fig. 4, if we want to find a subgroup of size two, users u_1 and u_2 will be returned as the desired subgroup as these two users with POIs (p_1^2, p_2^1) result in the subgroup with the minimum aggregate trip distance among all other subgroups of size two.

2.3 Group Trip Scheduling Queries

A group trip planning query computes a common trip for all group members, whereas a group trip scheduling query computes individual trips for the group members with an aim to minimize the aggregate trip distance. A group trip scheduling query is formally defined as follows:

Definition 4 ***(Group Trip Scheduling Queries).*** *Given a set of POI sets P, a group G of n users $\{u_1, u_2, \ldots, u_n\}$, a set of n source locations $S = \{s_1, s_2, ..., s_n\}$, a set of n corresponding destination locations $D = \{d_1, d_2, ..., d_n\}$ of the group members, a set of m POI types $T = \{t_1, t_2, ..., t_m\}$, and an aggregate function f, the group trip scheduling query returns n POI sets A_1, A_2, ..., A_n for $u_1, u_2, \ldots, u_n$, respectively that together minimizes the aggregate trip distance, a POI type in T is visited by a single group member, and n trips of group members together visits m POI types in T.*

If group members visit equal number of POI types then the group trip scheduling query is known as a uniform group trip scheduling query.

Figure 5 shows an example of a group trip scheduling query. In this example, seven POIs: three POIs p_1^1, p_1^2, p_1^3 of type 1 (i.e., Movie Theatres), two POIs

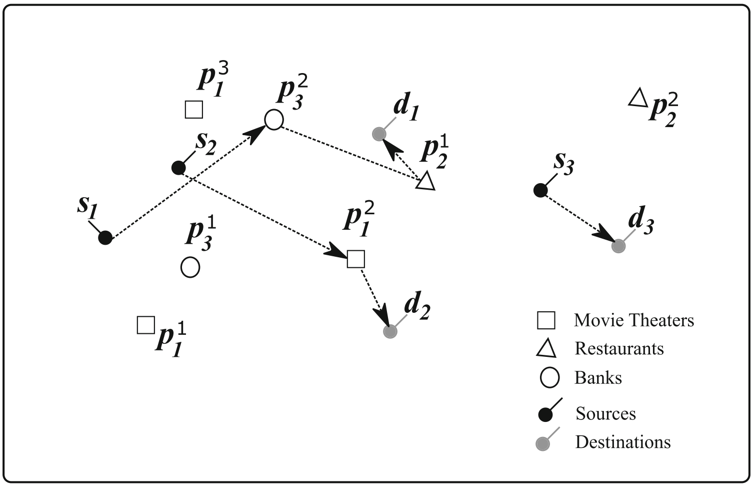


Fig. 5. A group trip scheduling query in an Euclidean space

p_2^1, p_2^2 of type 2 (i.e., Restaurants), two POIs p_3^1, p_3^2 of type 2 (i.e., Banks), and a group of three users with source locations s_1, s_2, s_3 and destination locations d_1, d_2, d_3 are shown. The group trip returns the schedule trip of each member, $s_1 \rightarrow p_3^2 \rightarrow p_2^1 \rightarrow d_1$, $s_2 \rightarrow p_1^2 \rightarrow d_2$, and $s_3 \rightarrow d_3$, which combinedly result in the minimum aggregate travel distance.

3 Overview of Existing Works

In this section, we first discuss existing works on trip planning queries and their variants (Sect. 3.1). Then we present existing works on group trip planning approaches (Sect. 3.2). After that we discuss two variants of group trip planning queries, namely subgroup trip planning queries (Sect. 3.3) and privacy preserving trip planning queries (Sect. 3.4). Finally we highlight the existing work on group trip scheduling queries (Sect. 3.5).

3.1 Trip Planning Queries

The trip planning query and its variants have been extensively studied in spatial databases due to its diversified applications in map based services [9,23,27,34]. The goal of the trip planning query is to find the best route (in terms of the minimum trip distance or the fastest travel time) that starts at the user's source location, goes through at least a POI of each type, and ends at the user's destination location. Li et al. [23] introduced the trip planning query in spatial databases, where they proposed four different approximation algorithms with various approximation ratios with respect to number of POI types and the number of POIs in each type. The proposed solutions work for both Euclidean and road network spaces.

A popular variation of the trip planning query is the optimal sequenced route, where the user gives her preferred order of visiting different types of POIs, e.g., first an ATM, then a coffee shop, and finally a bus stop. Sharifzadeh et al. [34] proposed the first solution for the optimal sequenced route query in spatial databases. The goal of the optimal sequenced route query is to find the best route (in terms of the minimum trip distance) that starts at the user's source location, goes through at least a POI of each type according to the specified order, and ends at the user's destination. Sharifzadeh et al. [34] developed techniques to solve the optimal sequenced route queries in both Euclidean and road network spaces. For the road network space, they have first proposed Dijkstraś algorithm based solution that works on the pre-computed shortest path distances from starting locations to all possible end points, which incurs large computational overhead. To overcome this limitation, they proposed an efficient technique, namely LORD, that first prunes a large number of POIs based on some derived threshold. Later, they proposed another technique, namely R-LORD, which converts the concept of threshold to range queries to retrieve a candidate set of POIs and find the best route. To solve the optimal sequenced route queries in Euclidean space, they proposed a technique that progressively find nearest neighbors to different points sets to construct the route.

Chen et al. [9] proposed a variant of the optimal sequenced route query, namely the multi-rule partial sequenced route (MRPSR) query. In the MRPSR query, a user can specify the order or a partial order of visiting POI of different types, and the goal is to find the best route satisfying those constraints. They proved that MRPSR is NP-hard and thus developed three heuristic algorithms to search for near-optimal solutions for the MRPSR query.

Ohsawa et al. [27] proposed an adaptation of incremental euclidean restriction algorithm that first retrieves POIs based on Euclidean distance to derive a bound, and then computes the network distances of POIs that fall inside the bound to compute the answers.

Aljubayrin et al. [4] introduced an interesting variant of trip planning queries, namely skyline trip queries, where the objective is to find the skyline trips in term of both trip length and the aggregated cost of the trip. The main difference of the trip planning query and the skyline trip query is that the trip planning query only finds the shortest trip, whereas the skyline trip query uses the trips distance and cost to find the set of skyline trips.

Cao et al. [7] proposed an interesting variant of the trip planning query, namely keyword aware optimal route query, that finds an optimal route covering a set of user specified keywords (e.g., "restaurant", "coffee", etc.) and satisfying a budget constraint of the user. Since the problem is NP-Hard, an approximation algorithm with probable approximation bound is proposed in this work.

Oshawa et al. [28] proposed the dynamic trip planning query that recomputes the trip if the user deviates from her original path of the trip. They proposed a safe-region based approach that avoids re-computation of a new trip if the user's deviation does not result in any change in the original trip.

3.2 Group Trip Planning Queries

Hashem et al. [17] have introduced group trip planning queries and proposed two algorithms: iterative and hierarchical, for evaluating group trip planning queries with an aim to minimize the total trip distance of the group members. The iterative algorithm repetitively constructs the trips for the group until the optimal trip is identified. The limitation of the iterative algorithm is that it accesses the same POIs multiple times in the database. The hierarchical algorithm addresses the limitation of the iterative algorithm and evaluates the answer with a single search on the database. Though the performance of the hierarchical algorithm improves than the iterative algorithm, still it cannot compute the trips for a large number of POI types (e.g., for more than 3 POI types) in a reasonable time. According to [18], the hierarchical algorithm requires 16 min to compute four best trips involving two POI types and 16 group members on commodity hardware, which is not acceptable. Another major limitation of [17] is that both algorithms assume that separate R-trees [6] to index POIs different types, which means a California dataset [1] that has 63 POI types requires 63 R-trees.

Hashem et al. [18] developed an efficient approach to process group trip planning queries in both Euclidean space and road networks, and consider both minimizing the total and the maximum trip distance of the group members. The authors refined the POI search space based on the elliptical properties and developed a dynamic programming technique to compute the trip for the group using the POIs in the refined search space. Specifically, the approach first computes a trip using a heuristic and finds the upper bound of the optimal aggregate trip distance of the group. Then the approach refines the POI search space as an ellipse with foci at the geometric centroids of the source and destination locations of the group members, respectively, and the length of the major axis equals to the upper bound of the average trip distance of the group members. It is guaranteed that a POI located outside the ellipse cannot be a part of the optimal answer.

In [18], the authors showed two techniques for retrieving the POIs inside the ellipse: range based retrieval and incremental retrieval. For the range based retrieval, the approach retrieves all POIs inside the ellipse and using the dynamic programming technique, computes the trip that provides the minimum aggregate trip distance for the group members. On the other hand, for the incremental retrieval, the approach incrementally retrieves the group nearest POIs with respect to the centroids of the source and destination locations of the group members. After retrieving a new POI, the approach computes the new possible trips and checks whether the new trips can improve the upper bound of the optimal aggregate trip distance. Thus, with the retrieval of a new POI, the ellipse remains same or shrinks. The algorithm terminates the search when the optimal trip is identified, i.e., all POIs inside the ellipse are retrieved.

Since the ellipse may become smaller with the incremental retrieval of POIs, the incremental retrieval technique accesses a smaller number of POIs than the range based algorithm. Experiments using real datasets show that the incremental algorithm is better than the range based retrieval technique and both range based and incremental retrieval techniques outperform the hierarchical algorithm [17] with a large margin. An additional advantage of [18] is that it uses a single R-tree to index the POIs in the database.

At the same time of [18], two other works [2,33] were developed to address the group trip planning queries. In [33], the authors refined the search space using the intersection of multiple ellipses, where each ellipse corresponds to a group member. The foci of the ellipse is at the source and destination locations of a group member and the length of the major axis equals to the upper bound of the aggregate trip distance of the group members, where the upper bound is determined using a heuristic. In [2], the authors developed an approach that finds the optimal trip for a group using the breadth and depth first search and applying pruning strategies.

Recently, Fan et al. [11] extended the group trip planning query that allows users to give preferences on POI types as well as the distance constraint, and finds a group trip that maximizes the social experiences by considering the agreement and disagreement of POI preferences among the group members. They have proposed both exact and approximate solutions for the above query.

Most recently, Tabassum et al. [36] proposed a new type of query, namely dynamic group trip planning query. The major difference between the group trip planning query and the dynamic group trip planning query is that in the traditional group trip planning query the group members remain static or fixed during the trip whereas in the dynamic group trip planning query members can join or leave after visiting any POI in the middle, where these changes of the group can be pre-determined or in real-time. They have developed a POI search refinement technique and a dynamic programming based solution to solve the dynamic group trip planning query.

3.3 Subgroup Trip Planning Queries

Hashem et al. [19] introduced a subgroup trip planning query that finds the subgroup and POIs from each required type having the minimum aggregate trip distance for any subgroup size. The motivation of the subgroup trip planning query comes from the following observation. In many real-world scenarios, it may happen that the majority of the users in the group are located close-by whereas a few members are located far away from them. In such a scenario, one may want to find a subgroup and the corresponding POI set if the aggregate trip distance of a subgroup improves significantly compared to the complete group. They maintain a separate R-tree for each type of POI, and take a best first strategy to retrieve POIs from each tree. A priority queue is maintained, where each entry in the priority queue consists of a tuple $(r_1, r_2, ...r_m)$, one item r_i (a POI/node) from each R-tree, and entries are sorted in ascending order of the minimum distance of the trip travelling through $r_1 \rightarrow r_2 \rightarrow ... \rightarrow r_m$. For each retrieved tuple, they compute the trip distance for each user separately going through these POIs, and adds the lowest n' (i.e., subgroup size) individual trip distances to get the subgroup with the minimum aggregate trip distances. Finally, we return the subgroup and the trip that result in the minimum aggregate trip distance among all possible subgroups of a given size. Note that, the solution of the subgroup trip planning queries has been adopted from the subgroup nearest neighbor queries presented in [3].

3.4 Privacy Preserving Group Trip Planning Queries

Soma et al. [35] proposed the first approach for protecting location privacy of users accessing trip planning queries. In the proposed work, a user can either provide a false location or cloaked locations (i.e., two regions including the actual source and destination locations of a user, respectively) to the service provider. For the false location, the user incrementally retrieves the nearest POIs with respect to the provided false location from the service provider until the POI set that minimizes the user's trip distance is identified. The authors developed a technique to identify the optimal answer with respect to the actual source and destination locations of the user by exploiting geometric properties. On the other hand, for cloaked locations, the service provider returns a candidate POI set that includes the optimal trips for all possible source and destination location combinations within the provided regions. Since the user knows her actual source and destination locations, the user can identify the actual answer from the candidate POI set. The authors developed an efficient algorithm to evaluate the candidate POI set with respect to the cloaked locations with a single search on the database.

3.5 Group Trip Scheduling Queries

A group trip scheduling query finds independent trips for the group members with an aim to minimize the aggregate trip distance of the group members. Thus neither the approaches for trip planning nor those for group trip planning are applicable for group trip scheduling queries. Jahan et al. [21] proposed the first approach for processing group trip planning queries. The overall steps to evaluate a group trip scheduling query is same as [18], i.e., computing the upper bound of the aggregate trip distance using a heuristic, refining the POI search space, and finding the trips using a dynamic programming technique. However, the ways the search space is Refined and the dynamic programming technique is developed are considering the constraints of a group trip scheduling query.

4 Comparative Analysis

In this section, we categorize and compare the existing works from different viewpoints. First, we categorize the works based on the number of users they support in the queries (Sect. 4.1). Then we divide the works based on the nature of data space, e.g., the Euclidean or road network spaces (Sect. 4.2). After that, we classify existing works on the requirements of the POI visiting sequence (Sect. 4.3). We also classify the works based on the static and dynamic nature of members in the group (Sect. 4.4). Later we divide the works based on the optimization criteria (Sect. 4.5). Finally, we present the existing work from privacy viewpoint (Sect. 4.6).

4.1 Single User vs. Groups

A trip planning query has been first introduced in [23] for a single user and since then a number of algorithms [4,9,27,34] have been developed to address the trip planing query and its variants for a single user. Later in [17], the authors proposed the first approach for processing group trip planning queries. Researchers have also focused on the performance improvement of the group trip planning and scheduling algorithms [2,18,21,33]. Recently, in [19], the authors developed an approach for planning trips by considering the subgroups from a group.

4.2 Space Type

In the literature, the proposed trip planning algorithms [9,17] addressed only the Euclidean space, whereas the algorithms proposed in [18,21,23,27] can also plan and schedule trips in the road networks. Only the approach proposed in [5] can plan trips for the pedestrians in the obstructed space, i.e., in presence of obstacles like a building, a fence or a pond.

4.3 Fixed vs. Flexible POI Sequence

The sequence of visiting different POI types can be fixed [27,33,34], partially fixed [9] and flexible [18,23]. For example, a user may want to visit a bank, a restaurant and a shopping mall. For a fixed sequence, a user can specify the order of visiting POI types: first restaurant, then a bank, a shopping mall at the end. For a partially fixed sequence, a user can specify that the user wants to visit the bank before visiting a shopping a mall; thus, three possible orders are: (i) bank, shopping mall, restaurant, (ii) bank, restaurant, shopping mall, (iii) restaurant, bank, shopping mall. For a flexible order, a user is happy to visit the POI types in any order.

4.4 Static vs. Dynamic

In static trip planing and scheduling queries, the query parameters and thus the answers do not change over time, whereas in dynamic queries, the query parameters and the answers may get updated with time. All existing works except [28,36] considered static settings. In continuous trip route planing queries [28], a user's deviation from the planned route may make the computed POI set that minimizes the trip distance for the user invalid; the query needs to be re-evaluated with respect to the user's changed location for identifying the POI set that minimizes the user's trip distance for the remaining part of the trip. In dynamic group trip planning queries [36], the group may change during the trip, i.e., a new member may join the trip at any time and existing member may leave. The change of group members may cause change in the POI set that minimizes the aggregate trip distance of the group members for the remaining part of the trip.

4.5 Optimization Parameters

Most of the existing approaches [17,18,21,23,34] considered distance as the optimization parameter for evaluating the trip planning and scheduling queries. Few other approaches consider popularity of routes [7], social preference of the group members [11], and cost of the trips [4] for planning and scheduling trips.

4.6 Privacy

Though location privacy issues have been extensively studied in the literature for range queries [20], shortest path queries [26], nearest neighbor queries [8,14,25], and group nearest neighbor queries [15,16], only in [35], the authors consider protecting location privacy of users while accessing trip planning queries. In [35], the authors developed techniques to compute the optimal answers for trip planning queries with respect to the false or cloaked locations of the users.

5 Conclusion and Future Works

In recent years, researchers have proposed algorithms for processing trip planning and scheduling queries and their variants. We have presented an overview of the variants of trip planning and scheduling queries that exist in the literature, and discussed the relevant state of the art techniques. We have performed a comparative analysis among the existing approaches based on different parameters that will help researchers to find the gaps and carry out further research in this area. Few open research directions for trip planning and scheduling queries are: (i) planning and scheduling trips with crowdsourced approaches [24] that do not need the presence of a service provider, (ii) introducing variants of group trip scheduling queries and developing solutions, (iii) investigating other privacy models like cryptography [12] and differential privacy [10] for protecting location privacy of users accessing trip planning and scheduling queries.

Acknowledgments. This research has been done in the department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET). The work is supported by the research grant from BUET.

References

1. California road network data (2017). https://www.cs.utah.edu/~lifeifei/SpatialDataset.htm
2. Ahmadi, E., Nascimento, M.A.: A mixed breadth-depth first search strategy for sequenced group trip planning queries. In: MDM, pp. 24–33 (2015)
3. Ali, M.E., Tanin, E., Scheuermann, P., Nutanong, S., Kulik, L.: Spatial consensus queries in a collaborative environment. ACM Trans. Spatial Algorithms Syst. **2**(1), 3:1–3:37 (2016)

4. Aljubayrin, S., He, Z., Zhang, R.: Skyline trips of multiple POIs categories. In: Renz, M., Shahabi, C., Zhou, X., Cheema, M.A. (eds.) DASFAA 2015. LNCS, vol. 9050, pp. 189–206. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18123-3_12
5. Anwar, A., Hashem, T.: Optimal obstructed sequenced route queries in spatial databases. In: EDBT, pp. 522–525 (2017)
6. Beckmann, N., Kriegel, H.P., Schneider, R., Seeger, B.: The R*-Tree: an efficient and robust access method for points and rectangles. In: SIGMOD, pp. 322–331 (1990)
7. Cao, X., Chen, L., Cong, G., Xiao, X.: Keyword-aware optimal route search. PVLDB **5**(11), 1136–1147 (2012)
8. Chao, I.M., Golden, B.L., Wasil, E.A.: "Don't trust anyone": privacy protection for location-based services. PMC **7**, 44–59 (2011)
9. Chen, H., Ku, W., Sun, M., Zimmermann, R.: The multi-rule partial sequenced route query. In: SIGSPATIAL, pp. 10:1–10:10 (2008)
10. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014)
11. Fan, L., Bonomi, L., Shahabi, C., Xiong, L.: Multi-user itinerary planning for optimal group preference. In: Gertz, M., et al. (eds.) SSTD 2017. LNCS, vol. 10411, pp. 3–23. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64367-0_1
12. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.: Private queries in location based services: anonymizers are not necessary. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, 10–12 June 2008, pp. 121–132 (2008)
13. Gutin, G., Karapetyan, D.: A memetic algorithm for the generalized traveling salesman problem. Nat. Comput. **9**(1), 47–60 (2010)
14. Hashem, T., Kulik, L.: Safeguarding location privacy in wireless ad-hoc networks. In: Krumm, J., Abowd, G.D., Seneviratne, A., Strang, T. (eds.) UbiComp 2007. LNCS, vol. 4717, pp. 372–390. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74853-3_22
15. Hashem, T., Kulik, L., Zhang, R.: Privacy preserving group nearest neighbor queries. In: EDBT, pp. 489–500 (2010)
16. Hashem, T., Ali, M.E., Kulik, L., Tanin, E., Quattrone, A.: Protecting privacy for group nearest neighbor queries with crowdsourced data and computing. In: UbiComp, pp. 559–562 (2013)
17. Hashem, T., Hashem, T., Ali, M.E., Kulik, L.: Group trip planning queries in spatial databases. In: Nascimento, M.A., Sellis, T., Cheng, R., Sander, J., Zheng, Y., Kriegel, H.-P., Renz, M., Sengstock, C. (eds.) SSTD 2013. LNCS, vol. 8098, pp. 259–276. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40235-7_15
18. Hashem, T., Barua, S., Ali, M.E., Kulik, L., Tanin, E.: Efficient computation of trips with friends and families. In: CIKM, pp. 931–940 (2015)
19. Hashem, T., Hashem, T., Ali, M.E., Kulik, L., Tanin, E.: Trip planning queries for subgroups in spatial databases. In: Cheema, M.A., Zhang, W., Chang, L. (eds.) ADC 2016. LNCS, vol. 9877, pp. 110–122. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46922-5_9
20. Hu, H., Lee, D.L.: Range nearest-neighbor query. IEEE Trans. Knowl. Data Eng. **18**(1), 78–91 (2006)
21. Jahan, R., Hashem, T., Barua, S.: Group trip scheduling (GTS) queries in spatial databases. In: EDBT, pp. 390–401 (2017)

22. Laporte, G.: A concise guide to the traveling salesman problem. JORS **61**(1), 35–40 (2010)
23. Li, F., Cheng, D., Hadjieleftheriou, M., Kollios, G., Teng, S.-H.: On trip planning queries in spatial databases. In: Bauzer Medeiros, C., Egenhofer, M.J., Bertino, E. (eds.) SSTD 2005. LNCS, vol. 3633, pp. 273–290. Springer, Heidelberg (2005). https://doi.org/10.1007/11535331_16
24. Mahin, M.T., Hashem, T., Kabir, S.: A crowd enabled approach for processing nearest neighbor and range queries in incomplete databases with accuracy guarantee. Pervasive Mob. Comput. **39**, 249–266 (2017)
25. Mokbel, M.F., Chow, C., Aref, W.G.: The new casper: a privacy-aware location-based database server. In: ICDE, pp. 1499–1500 (2007)
26. Mouratidis, K., Yiu, M.L.: Shortest path computation with no information leakage. PVLDB **5**(8), 692–703 (2012)
27. Ohsawa, Y., Htoo, H., Sonehara, N., Sakauchi, M.: Sequenced route query in road network distance based on incremental Euclidean restriction. In: Liddle, S.W., Schewe, K.-D., Tjoa, A.M., Zhou, X. (eds.) DEXA 2012. LNCS, vol. 7446, pp. 484–491. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32600-4_36
28. Ohsawa, Y., Htoo, H., Win, T.N.: Continuous trip route planning queries. In: Pokorný, J., Ivanović, M., Thalheim, B., Šaloun, P. (eds.) ADBIS 2016. LNCS, vol. 9809, pp. 198–211. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44039-2_14
29. Papadias, D., Shen, Q., Tao, Y., Mouratidis, K.: Group nearest neighbor queries. In: ICDE, pp. 301–310 (2004)
30. Papadias, D., Tao, Y., Mouratidis, K., Hui, C.K.: Aggregate nearest neighbor queries in spatial databases. ACM Trans. Database Syst. **30**(2), 529–576 (2005)
31. Rego, C., Gamboa, D., Glover, F., Osterman, C.: Traveling salesman problem heuristics: leading methods, implementations and latest advances. Eur. J. Oper. Res. **211**(3), 427–441 (2011)
32. Roussopoulos, N., Kelley, S., Vincent, F.: Nearest neighbor queries. In: SIGMOD, pp. 71–79 (1995)
33. Samrose, S., Hashem, T., Barua, S., Ali, M.E., Uddin, M.H., Mahmud, M.I.: Efficient computation of group optimal sequenced routes in road networks. In: MDM, pp. 122–127 (2015)
34. Sharifzadeh, M., Kolahdouzan, M.R., Shahabi, C.: The optimal sequenced route query. VLDB J. **17**(4), 765–787 (2008)
35. Soma, S.C., Hashem, T., Cheema, M.A., Samrose, S.: Trip planning queries with location privacy in spatial databases. World Wide Web **20**(2), 205–236 (2017)
36. Tabassum, A., Barua, S., Hashem, T., Chowdhury, T.: Dynamic group trip planning queries in spatial databases. In: SSDBM, pp. 38:1–38:6 (2017)
37. Xu, Z., Rodrigues, B.: A 3/2-approximation algorithm for multiple depot multiple traveling salesman problem. In: Kaplan, H. (ed.) SWAT 2010. LNCS, vol. 6139, pp. 127–138. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13731-0_13
38. Yiu, M.L., Mamoulis, N., Papadias, D.: Aggregate nearest neighbor queries in road networks. IEEE Trans. Knowl. Data Eng. **17**(6), 820–833 (2005)

Precision/Recall Trade-Off Analysis in Abnormal/Normal Heart Sound Classification

Jeevith Bopaiah[2] and Ramakanth Kavuluru[1,2](✉)

[1] Division of Biomedical Informatics, Department of Internal Medicine,
University of Kentucky, Lexington, KY, USA
[2] Department of Computer Science, University of Kentucky, Lexington, KY, USA
{jeevith.bopaiah,ramakanth.kavuluru}@uky.edu

Abstract. Heart sound analysis is a preliminary procedure performed by a physician and involves examining the heart beats to detect the symptoms of cardiovascular diseases (CVDs). With recent developments in clinical science and the availability of devices to capture heart beats, researchers are now exploring the possibility of a machine assisted heart sound analysis system that can augment the clinical expertise of the physician in early detection of CVD. In this paper, we study the application of machine learning algorithms in classifying abnormal/normal heart sounds based on the short ($\leq$120 s) audio phonocardiogram (PCG) recordings. To this end, we use the largest public audio PCG dataset released as part of the *2016 PhysioNet/Cardiology in Computing Challenge*. The data comes from different patients, most of who have had no previous history of cardiac disease and some with known cardiac diseases. In our study, we use these audio recordings to train three different classification algorithms and discuss the effects of class imbalance (normal vs. abnormal) on the precision-recall trade-off of the prediction task. Specifically, our goal is to find a suitable model that takes into account the inherent imbalance and optimize the precision-recall trade-off with a higher emphasis on increasing recall. Bagged random forest models with majority (normal) class under sampling gave us the best configuration resulting in average recall over 91% with nearly 64% average precision.

1 Introduction

For the past decade, cardiovascular diseases (CVD) have been the leading cause of deaths around the globe. According to the WHO statistics, as of 2015, ischemic heart disease is the "world's biggest killer" (http://www.who.int/mediacentre/factsheets/fs310/en/). According to a report published in 2017 by American Heart Association, CVD accounts for 801,000 deaths in the United States [1]. Most of these deaths could be prevented if the diseases were detected in their early stage. Auscultation is a procedure used by the physicians to examine the heart. It involves listening to the heart sounds to detect abnormality in the heart. This requires substantial experience and is a complex process prone to human error. Also, the patient to doctor ratios are extremely high in certain parts of

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 179–194, 2017.
https://doi.org/10.1007/978-3-319-72413-3_12

the world (up to tens of thousands) and hence manual examination is not ideal in many cases. Given these situations, cloud based solutions that allow more accurate preliminary examination of heart health based on heart sounds may offer an important alternative. Central to such a service would be a high quality predictive model that can identify abnormal heart sounds automatically. To make this a reality, researchers around the world are building expert annotated datasets and machine learned models. The *2016 PhysioNet/Computing in Cardiology Challenge (CinC)* [5] provided the largest public heart sound database with which researchers built supervised models and tested against a hidden test set. Although the competition ended late 2016, the hidden test set has not been made public yet. In this paper, we study the efficacy of classical machine learning algorithms in identifying abnormal heart sounds with a focus on the precision-recall trade-off. Before we proceed, we first discuss how heart sounds are generated and measured.

Heart sounds are produced by four distinct events that take place in the heart. These four events correspond to the mechanical activity of opening and closing of the valves in the heart. Each heart beat is triggered by an electrical impulse inside the heart that causes the atrium and ventricles to contract and relax alternatively [4]. This consecutive contraction and relaxation event draws impure blood into the heart and pumps out pure blood to the rest of the body. Each heart cycle is composed of these four events that occur in quick succession in a particular order. The actual sequence of events is *S1, systole, S2*, and *diastole* where S1 and S2 correspond to the fundamental sounds made by the heart via its mechanical movements. Along with these, the heart recordings may also contain other sounds such as systolic ejection click (EC), mid-systolic click (MC), the diastolic sound (OS), as well as heart murmurs caused by the flow of blood [8]. All these sounds can be captured using a phonocardiograph which produces an audio file. The audio recording should at least be long enough to contain an entire heart cycle. In this project our task involves developing a predictive model that analyzes the sound patterns of the audio file to predict the corresponding heart beat as either normal or abnormal. This allows more accessible, real time monitoring of the heart that can be used to assist physicians in preliminary checks for CVDs.

Given this motivation, researchers have been working in the field of heart sound analysis for the past five decades but most of their efforts have had drawbacks in terms of access to very few heart sound recordings, lack of a separate test dataset, and failure to use a variety of PCG recordings [5]. However, these drawbacks have been mitigated with the introduction of the *2016 PhysioNet/CinC Challenge* dataset. Some of the recent works [11,12,17] on this dataset include the use of deep neural networks and ensemble approaches (more details in Sect. 8). However, most of these efforts do not analyze the trade-off between recall and precision. Actually, they all analyze accuracy which is defined for them as the simple mean of recall and specificity (which is different from precision). However, for classification tasks with imbalanced datasets where the minority class is the positive class that is of interest, it is well known that precision and recall are more important [13]. Our effort is focused on precision-recall analysis while also disclosing accuracy information.

2 Dataset

The dataset [6,8] used in our experiments was obtained from a publicly available heart sound database which was hosted by the *PhysioNet* group. This dataset was compiled by various researchers around the world who have collected eight heart sound databases, each sourced from different healthcare facilities and home visits. These heart sounds were recorded with a sampling rate of 44 kHz which was then downsampled to 2000 Hz. Out of these eight databases, six were made publicly available for training the models while the remaining two databases along with few records from training dataset were kept private as blind test data. The summary of the training dataset is shown in Table 1.

Table 1. Physionet/CinC challenge training dataset summary [8]

Database Name	# Raw Recordings		
	Abnormal	Normal	Total
Database-a	292	117	409
Database-b	104	386	490
Database-c	24	7	31
Database-d	28	27	55
Database-e	183	1958	2141
Database-f	34	80	114
Total	665	2575	3240

The public dataset consists a total of 3,240 heart sound recordings. The length of each recording varies between 5 and 120 s. The average length of heart cycle within each recording is 1.5–2 s. Thus, each recording is long enough to contain more than one heart cycles. Typically, higher number of heart cycles present in a recording allows for a better representation of abnormal patterns during feature extraction. This is analogous to the fact that learning algorithms generalize better with more relevant data points. The entire dataset has around 80,000 heart cycles in it. These recordings can be parsed to produce a vector of amplitudes varying in time.

3 Methods

The architecture of our predictive model for the heart sound classification task is shown in Fig. 1. In this approach, we experiment with three well known classification algorithms: random forests, logistic regression, and support vector machines

(SVM). Of the three algorithms, we found random forest algorithm to be more effective in classifying the heart beats as either normal or abnormal if F-score is the chosen measure[1]. Random forests are an ensemble model formed by employing multiple decision trees as base classifiers. Each tree in the random forest is trained on a randomized smaller subset of the full set of features. These individual trees behave as weak learners with complementary characteristics and hence are combined to create a powerful learning algorithm that uses a voting mechanism among the different trees to obtain the final predictions. Furthermore, we have optimized the hyper-parameters: *number of trees* and *depth of each tree* by using an exhaustive search algorithm. We found the best configuration involved 200 trees with a tree depth of up to 20 levels.

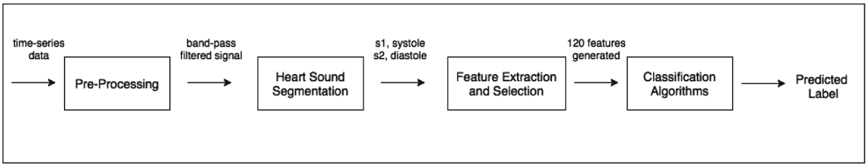


Fig. 1. Predictive modeling pipeline used in heart sound classification

The different stages involved in our predictive model are as follows.

3.1 Pre-processing

At the time of recording the heart sound, noises from the external environment or internal body functions are recorded along with the actual heart sounds. These background noises distort the actual signal and have a negative influence on the final predictions. The pre-processing stage involves de-noising the signal to contain only the actual heart sounds. The input signal, in the form of time series values of amplitudes, is further downsampled to 1000 Hz [11]. Downsampling is a process in which we reduce the number of data points/second in the input signal. The pre-processed signal now consists of 1000 amplitude values per second. This is useful especially when the sampling rate is much larger than the highest frequency component of the signal and processing the data becomes a challenge. From literature [8], we know that the heart sounds lie in the frequency range of 25 Hz–500 Hz. According to the Nyquist sampling theorem [3], there is no information loss if the sampling rate is at least twice the highest frequency component of the signal. Since we know that the highest desired frequency component is 500 Hz, we can downsample the signal to 1000 Hz without much loss of information. In the next step, we pass the signal through band-pass filters module that retains the frequencies in the range 25 Hz–500 Hz and eliminates the

[1] Henceforth, we only discuss the results using the random forest approach. Comparisons with the other two classifiers are presented in Sect. 6.

rest. This helps to weed out undesired frequencies less than 25 Hz and greater than 500 Hz. The signal is passed through a spike removal process that removes sharp peaks. The spike removal process helps in removing the noises from the external environment that appear as spikes in the signal. Finally, the signal is normalized to reduce the effect of extremely large or small amplitudes.

3.2 Heart Sound Segmentation

In this stage, each heart sound recording is segmented into four distinct heart sounds: S1, systole, S2, diastole. Each of the four heart sounds exhibits a distinct waveform pattern. Any variation in one or more of these sounds could potentially indicate an abnormality. Segmenting the entire heart sound recording helps in analyzing each of these four heart sounds for abnormal patterns. As suggested by the organizers of the 2016 PhysioNet challenge, we used the available state of the art segmentation algorithm developed by Springer et al. [14]. They use ECG as a reference signal to identify the approximate locations of the four heart sounds on a PCG signal. The PCG signal corresponds to the actual heart sound recording. Figure 2 shows the segmented PCG signal along with the ECG waveform.

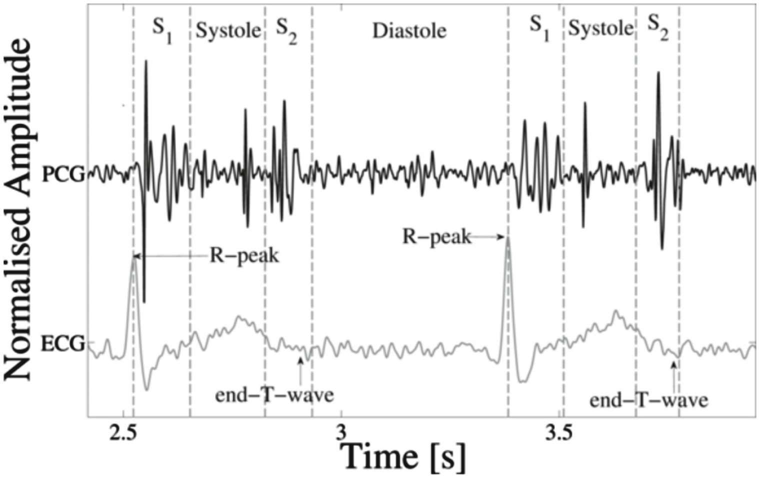


Fig. 2. PCG and ECG waveforms (from Liu et al. [8])

ECG measures the electrical impulse of the heart and is less prone to noise than a PCG signal. However, recording ECG is an expensive process and it is recommended by the physician only if needed at a later stage. The R-peaks (shown in Fig. 2) of the ECG coincide with the S1 phase of the heart beat. Similarly, the end-T-wave of the ECG coincides with the end of the S2 phase. Thus, using the R-peaks and end-T-waves of the ECG, the location of the heart sounds on a PCG are identified. The segmentation algorithm uses logistic regression coupled with a hidden semi-Markov model to predict the most likely sequence of states for each recording. The hidden semi-Markov model maximizes the likelihood of each data point to be in one of the four states while the logistic regression classifier models the expected duration densities for each state.

3.3 Feature Extraction and Selection

Based on the boundary regions of S1 identified in the segmentation step, we divide the entire heart sound recording into individual heart cycles. The features are extracted from each heart cycle and then averaged across the other heart cycles in the recording[2]. The features extracted from each heart cycle can be classified into two feature classes: time domain features and frequency domain features. The time domain features are comprised of the aggregate measures of the heart sound states. They can be further categorized into PCG intervals and PCG amplitudes.

The PCG intervals measure the time intervals of the various components of the heart recording. Features from the PCG intervals include mean and standard deviation of the following:

1. Length of the heart cycle
2. S1 interval length
3. Systole interval length
4. S2 interval length
5. Diastole interval length
6. Ratio of length of the systolic interval to the length of the heart cycle
7. Ratio of length of the diastolic interval to length of the heart cycle
8. Ratio of length of the systolic interval to that of the diastolic interval

The PCG amplitudes measure the aggregates of the amplitude values in the signal. These include the mean and standard deviation of the following:

1. Ratio of the mean amplitude in systole to the mean amplitude in S1
2. Ratio of the mean amplitude in diastole to the mean amplitude in S2
3. Skewness and kurtosis of amplitude in S1
4. Skewness and kurtosis of amplitude in systole
5. Skewness and kurtosis of amplitude in S2
6. Skewness and kurtosis of amplitude in diastole

Thus 36 features have been extracted from the time domain signal. The remaining 84 features were obtained from the frequency domain signal by using acoustic properties of the sound waves [11]. These include

1. Power Spectral Density
2. Mel-Frequency Cepstral Coefficients.

[2] We emphasize that prediction of abnormality is made per recording, not per cycle, given a full recording's multiple cycles together provide the signal for prediction.

Power Spectral Density (PSD) [9, Chap. 11]**:** It refers to the variances in amplitude in terms of the frequency of the signal. In simple terms, it measures the distribution of energy over the various frequency components of the signal. In order to compute the PSD, we first extract the frequency components that exists in a signal. Thus, the input time domain signal needs to be transformed into frequency domain signal. Fast Fourier transform is a signal processing technique that converts a time domain signal to its frequency domain. The PSD is measured for each of the four heart sounds: S1, systole, S2, diastole across nine different frequency bands: 25–45 Hz, 45–65 Hz, 65–85 Hz, 85–105 Hz, 105–125 Hz, 125–150 Hz, 150–200 Hz, 200–300 Hz, 300–400 Hz. This gives us a vector of 9 values for each of the four types of sounds and a total of 36 features for each heart cycle.

Mel-Frequency Cepstral Coefficients (MFCC): MFCC [7] is a powerful transformation technique that is popular among the speech recognition enthusiasts. It is based on the premise that humans perceive sound on a non-linear scale. In other words, the relationship between energy present in the sound and the loudness perceived by the human ear is non-linear as we transition from lower frequencies to higher frequencies. Increasing the intensity of a sound by a factor X, does not increase the loudness we hear by the same factor X. This is especially true for higher frequencies, where two sounds of frequencies, say for example, 4000 Hz and 4500 Hz are indistinguishable to the human ear. This non-linear relationship between the perception of the sound versus the actual energy present in the sound is modeled on a scale known as the mel scale. MFCC is an extension of the power spectral density graph in which the frequency in hertz is converted into frequency in mel using the below formula [7]

$$mel(f) = 2595 * \log_{10}\left(1 + \frac{f}{700}\right).$$

These frequencies in mels are used to create triangular filters that capture the energy present within each filter. Discrete cosine transformations are applied to the energies obtained from the mel filters to obtain the MFCC. The number of coefficients correspond to the number of filters. We have used 12 filters for each heart sound and each coefficient is considered as a feature. Thus, we have a total of $12 \times 4 = 48$ features.

Combining all the features we now have 120 features that can be used in training the random forest classifier. For feature selection, we used random forest classifier to identify a set of 81 informative features that determine the classification of the heart sound. Feature importance in random forests is determined by ranking the features based on its information gain. In each of the constituent decision trees, the feature chosen at each node is the one that maximizes the information gain at that node. Thus based on the ‘gini impurity’ (GI) measure, the features that maximizes the information gain across the different decision trees are ranked higher in the feature selection list [2]. We use GI because it is obtained as a direct consequence of using the random forest classifier and closely relates to the classifier’s underlying principle. Specifically, a feature with low GI score is more desirable to the classifier than a feature with high GI score. Since random forest classifier also takes into consideration the node impurities while predicting the label, GI appears to be a more appropriate feature selection criterion. The feature importance scores are normalized across all the features. By experimenting with thresholds of 0.006, 0.005, 0.004, 0.003 on feature importances, it was observed that a threshold of 0.005 resulted in 81 features that produced the best results. Among the 81 features, we found 2 feature classes (shown in Table 2) that were more prominent than the rest.

Table 2. Prominent features of the random forest classifier

Feature Class	Feature Score
Mel Frequency Cepstral Coefficients of Diastole Region	0.069
Mel Frequency Cepstral Coefficients of Systole Region	0.027

3.4 Random Forest Classifier Configurations

The 81 features obtained from feature selection process were used in training a random forest classifier. Initially, we used all the 3240 samples in training the classifier and noticed that the recall was averaging around 70%. Since the objective of this classification problem is to maximize recall without making prohibitive compromises on precision, we have implemented different configurations to study the effects of majority class under sampling on recall. These configurations are constructed by retaining all the samples from the minority class and varying the proportions of the majority class. On analyzing the results of these different configurations, we noticed that as the imbalance between the two classes decreased, the recall improves up to a certain threshold, beyond which it results in a loss of precision. In order to demonstrate this effect, we describe the last four configurations that capture the shift from increase in recall to decrease in precision.

To overcome class imbalance, we under sample the majority class and use a bagging approach on different bootstrap samples [15]. The dataset is split into 90% training and 10% test sets with train size of 2925 samples and test size of 315 samples. The test proportions of the positive and negative samples have been retained as in the original dataset (roughly 20% positive and 80% negative). Given we only under sample majority class, the minority class count is always the same (specifically, from Table 1 we have $665 \cdot 0.9 \approx 600$).

- **Model Configuration 1**: In this configuration, the number of positive (abnormal) examples is kept constant at 600 and the negative sample size is varied in increments of 100 (so 600, 700, ..., 2200) for each model. Thus we have 17 different classifiers. For each negative sample size, we train ten classifiers, for each of which the negative examples are chosen without replacement so that there are no duplicates. The final prediction is based on a voting mechanism with equal contribution from each of the 170 classifiers.

- **Model Configuration 2**: This configuration is a subset of the first configuration. We choose 600 positive examples and 900 negative examples. We train ten models with random sampling on the negative examples. The final prediction is based on the voting with the ten models.
- **Model Configuration 3**: This is similar to configuration 2 except that the number of negative examples is decreased to 800.
- **Model Configuration 4**: This is also similar to the second configuration with the number of negative examples further reduced to 700.

Given there are an even number of classifiers, ties are broken in favor of the minority class.

Evaluation Strategy: The 2016 PhysioNet challenge organizers use recall (also called sensitivity) and specificity metrics where specificity is the ratio of the true negatives to the sum of true negatives and false positives. For this particular task, the proposed recall and specificity metrics depend on specific weights determined by the number of 'noisy' and 'clean' records. Unfortunately, we do not have access to the noisy/clean labels for the public database; they were only provided for the hidden test set that is still not made public. We believe that precision, recall, and F-score are more informative for this task with a minority positive class of interest. For this, a realistic evaluation of the predictive model should account for the prevalence among the two classes. The area under the receiver operating characteristic (ROC) curve, representing a trade-off between recall and specificity, is shown to overestimate the performance of the model in imbalanced datasets with a minority positive class [10]. Hence, we chose the precision/recall as the main metrics that take into account the prevalence of the disease while evaluating the performance of the predictive model.

4 Results and Discussion

The results of the four different configurations are shown in Table 3. A quick glance at the results, especially the F-score and accuracy[3] may appear to be more or less similar in all the four configurations. This is also true to some extent until we take into considerations the key metrics: precision and recall. Though we could maximize any of the evaluation metric listed above, the one more suited for this task is maximizing the recall without incurring prohibitive losses in precision. A more fine grained observation reveals that the recall measure improves as the number of negative samples decreases across the different configurations. In the second configuration, we choose 900 negative samples. This was based on our experiments which showed that the recall drops significantly if the number of negative samples is beyond 900. Similarly, when the negative samples are reduced to below 700, the drop in precision is greater than the improvement in the recall.

[3] The notion of accuracy used here is the same as in the 2016 CinC challenge where it is set to (recall+specificity)/2.

Table 3. Random forest classifier performance measures

	Precision	Recall	F-score	Accuracy
Model Configuration 1	0.778	0.754	0.766	0.849
Model Configuration 2	0.697	0.815	0.752	0.862
Model Configuration 3	0.626	0.877	0.731	0.870
Model Configuration 4	0.615	0.908	0.733	0.879

Table 4. Average results of Config 4 via experiments with 40 distinct train-test splits

	Precision	Recall	F-score	Accuracy
Model Configuration 4	0.637	0.912	0.749	0.887

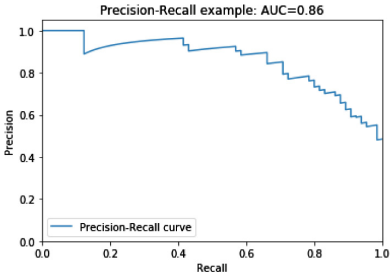


Fig. 3. Model Configuration 1

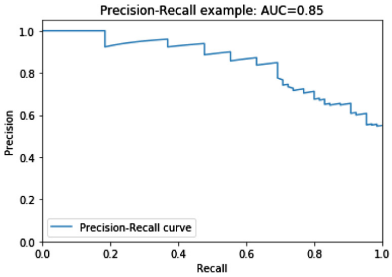


Fig. 5. Model Configuration 2

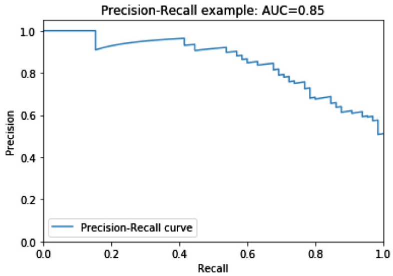


Fig. 4. Model Configuration 3

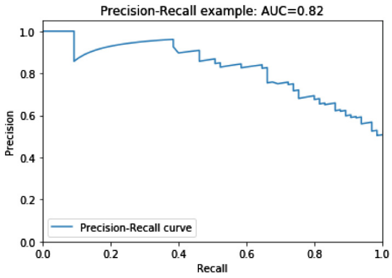


Fig. 6. Model Configuration 4

With configuration 4, we achieve a recall of 0.908 and precision of 0.61. This means that it could catch over 90% of the patients with cardiovascular diseases with precision of 61% – implying 39% of cases predicted as abnormal are actually normal. Even though the numerical value of precision makes the classifier appear very poor, for practical purposes this is not really a major hurdle. Specifically, given the number of instances predicted to be of the minority class is very low compared with the number predicted for the majority class, the manual burden of weeding out these additional healthy cases is also low given the 39% proportion is out of instances predicted to be abnormal.

In order to assess the stability of the results from configuration 4, we repeated the experiment 40 times by considering a different train-test split each time. The average results of the 40 runs are shown in Table 4. These results are similar to those in Table 3. To demonstrate this, we establish confidence intervals on the results obtained from the 40 runs. At 95% confidence, the accuracy is shown to be within 0.888 ± 0.0068 The tight bounds on the accuracy show that the performance is expected to generalize well.

The precision-recall (PR) curves for the four configurations are shown Figs. 3, 4, 5 and 6. As we can see, the area under the PR curves (AUPRC) is similar in all configurations but is slightly lower in the 4th configuration at 0.82, which is around four points lower compared with the first configuration. However, it is also clear (as we conveyed earlier) from a practical perspective, configuration 4 is more desirable.

5 Error Analysis

From the results of the random forest classifier, we know that the model suffers from a low precision score. To analyze the classification errors, we provide our error analysis on one of the 40 runs we conducted to generate results in Table 4. The prediction results, in terms of true positives, true negatives, false positives, and false negatives, are shown as a confusion matrix in Fig. 7.

The confusion matrix indicates an error of 14.4% false positives and 9.2% false negatives. On analyzing the euclidean distance between the feature vectors of the training samples and the misclassified test samples, we found that a significant portion of the test instances were closer to their incorrectly predicted class than their true class. Thus feature characteristics caused some of the samples to be misclassified. Specifically, Table 5 shows the percentages of false positive and false negative errors that are similar to positive and negative classes, respectively.

Table 5. Percentage of the test errors that are similar to the true classes

	Closer to negative samples	Closer to positive samples
False Positives	38.8%	61.1%
False Negatives	83.33%	16.66%

61.1% of test errors that were incorrectly predicted as abnormal, were closer to the abnormal training samples on average. Similarly, 83.33% of test errors that were incorrectly predicted as normal were closer to the normal training samples. It is clear that the boundary case counts are non-trivial and additional features that are more discriminative may be needed to improve the performance.

The numerical distribution of the errors across different databases (subsets of the dataset originating from different labs) is shown in Table 6. The databases are arranged in the increasing order of the sample size with database-c having the least number of samples and database-e having the highest sample size. From Table 6, we can observe that the test error decreases as the samples size increases with the exceptions of database-b and database-f. The percentage error shows that except for database-e, all the other databases perform poorly in classifying the heart sounds. On examining the original distribution of heart sounds among different databases, the correlation between the percentage error and the sample size in each database is apparent. In the original dataset, database-e has the

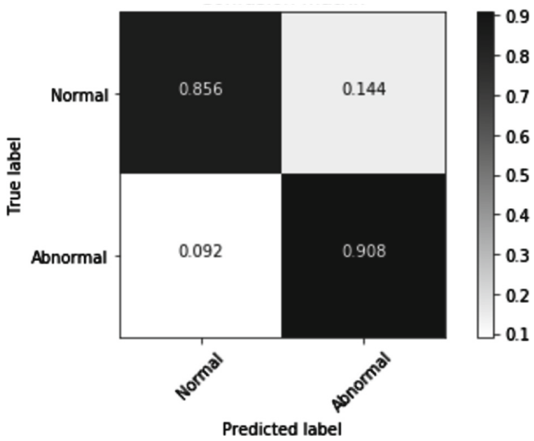


Fig. 7. Confusion matrix on one of the sample runs of our model

Table 6. Test error distribution among different subsets

Database Name	% Data distribution	% Test error
Database-c	0.95	33.33
Database-d	1.69	40.00
Database-f	3.51	52.94
Database-a	12.6	23.68
Database-b	15.12	44.44
Database-e	66.08	0.5

maximum number of heart sound recordings and it decreases with databases b, a, and f, to an extent that databases d and c have only 55 and 31 heart sound recordings respectively. Thus, we have many errors for databases which have fewer samples and few errors for database-e which has the highest number of heart sound recordings. As mentioned earlier, these databases are obtained from different healthcare facilities in which the recording instruments and locations of recording are different. Since the pattern of error in the instruments and the surrounding environment might be different for different healthcare centers, a model trained on only one particular database is more likely to perform poorly on the other. To build a more generalized model that performs well with data from different sources, the model should be trained on larger datasets from each of these sources. This would help capture the variations present in the data from different sources and should generalize well on a variety of heart sounds.

6 Comparison: Random Forest vs Other Classifiers

Apart from the random forest classifier, we have also explored two other classification algorithms: SVMs and logistic regression. The experimental settings were using the configuration 4 from Sect. 3.4 in terms of the majority class under sampling. Hyper parameters were fine-tuned using grid search. For this particular task, we found that SVMs are biased towards positive/abnormal class and more instances are predicted as abnormal thus resulting in better recall and lower precision. The loss in precision is nearly proportion to the gain in recall. As such, further exploration might be warranted in the future. Logistic regression also suffers from the same issue as with SVMs but the situation is much worse in terms of loss in precision. The results of these classifiers are shown in Table 7.

Table 7. Comparison: Random forest, SVM and logistic regression

	Precision	Recall	F-score	Accuracy
Random Forest Classifier	0.637	0.912	0.749	0.887
Support Vector Machines	0.581	0.959	0.722	0.889
Logistic Regression	0.462	0.965	0.624	0.836

7 Limitations

Although our effort shed light on the precision-recall trade-off aspects in heart sound classification, we have the following limitations.

– We still do not have public access to the hidden test set that was actually used for evaluation during the 2016 PhysioNet/CinC challenge. Hence a direct comparison of our results against challenge participants is not possible. The metric used is also different based on weights given to noisy examples. However, our accuracy of 88.7 in Table 4 is on par with other researchers' [16,17] cross-validation experiments[4] on the public training data. Furthermore, our parameter tuning was focused on the objective of maximizing F-score (not accuracy) suitable for situations with class imbalance with minority positive class.
– Our model requires that the heart recording be long enough to have at least 2–3 heart cycles in it as the model generalizes well with more number of heart cycles, improving the accuracy of the system.
– Since there are various types of cardiovascular diseases, it is quite possible that the training samples are not representative of all the cardiac diseases.

8 Related Work

Here we outline prior efforts from the 2016 PhysioNet/CinC challenge participants. Potes et al. [11] employed the aggregate features we used in Sect. 3.3 to train a AdaBoost-abstain classifier composed of several weak learners, one for each feature. They also used four convolutional neural networks (CNNs) on each heart cycle, one for each of frequency ranges 25–45 Hz, 45–80 Hz, 80–200 Hz, and 200–400 Hz. The output of these four CNNs is flattened and input to a multi-layer perceptron. The final decision is made using a combination of the AdaBoost and CNN models. They have achieved recall of 94.24% and specificity of 77.81%. Rubin et al. [12] used the spectral features such as MFCC to obtain a two-dimensional time-frequency heat map representation. This 2-D heat map is used in training a deep convolutional neural network. With this approach they have achieved a high specificity of 93.1% and a low recall rate of 76.5%. Zabihi et al. [17] avoid the heart sound segmentation phase by using an ensemble of 20 feed forward neural networks to predict the final result by a voting mechanism. They used features based on the properties of the sound waves, extracted from time domain, frequency domain, and time-frequency domain signals to transform the input signal to a more meaningful representation before feeding it to the neural network. Although they avoided the segmentation process, they obtained comparable results with a specificity of 84.9% and recall of 86.9%.

9 Conclusion

In this paper, we present the details of supervised heart sound classification experiments we conducted using the 2016 PhysioNet/CinC challenge. Using random forests, SVMs, and logistic regression, we showed that a recall over 90% can be achieved and specifically using bagged random forests with under sampling we

[4] Even this may not be exact comparison because the numbers of folds were different.

show that this can be done with a precision of 64%. Most of the features we used are inspired by the efforts in the signal processing community. However, based on error analysis experiments, we conclude that a richer feature space might be needed to build better models especially in terms of increasing precision. As a next step, we could explore more complex ensembles using a wide variety of classification algorithms (including deep neural networks) to improve precision. Another area to be explored is to find the right combination of signal processing techniques that projects the input signal to a different feature space where the patterns are more clearly distinguishable. With more people working in this field and better performing systems, real time monitoring of the heart health could enable early detection of cardiovascular disease in low resource settings and decrease the mortality due to this disease.

Acknowledgements. We thank anonymous reviewers for their honest and constructive criticism of our paper. Our work is primarily supported by the National Library of Medicine through grant R21LM012274. We are also supported by the National Center for Advancing Translational Sciences through grant UL1TR001998. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

References

1. American Heart Association: Heart disease and stroke statistics (2017). At-a-glance. https://www.heart.org/idc/groups/ahamah-public/@wcm/@sop/@smd/documents/downloadable/ucm_491265.pdf
2. Archer, K.J., Kimes, R.V.: Empirical characterization of random forest variable importance measures. Comput. Stat. Data Anal. **52**(4), 2249–2260 (2008)
3. Olshausen, B.A.: Aliasing. http://redwood.berkeley.edu/bruno/npb261/aliasing.pdf
4. Cleveland Clinic: Heart and blood vessels: how does the heart beat. https://my.clevelandclinic.org/health/articles/heart-blood-vessels-heart-beat
5. Clifford, G.D., Liu, C., Moody, B., Springer, D., Silva, I., Li, Q., Mark, R.G.: Classification of normal/abnormal heart sound recordings: the physionet/computing in cardiology challenge 2016. In: Computing in Cardiology Conference (CinC), pp. 609–612. IEEE (2016)
6. Goldberger, A.L., Amaral, L.A., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.-K., Stanley, H.E.: Physiobank, physiotoolkit, and physionet. Circulation **101**(23), e215–e220 (2000)
7. Hasan, M.R., Jamil, M., Rabbani, M.G., Rahman, M.S.: Speaker identification using mel frequency cepstral coefficients. In: 3rd International Conference on Electrical and Computer Engineering, pp. 565–568 (2004)
8. Liu, C., Springer, D., Li, Q., Moody, B., Juan, R.A., Chorro, F.J., Castells, F., Roig, J.M., Silva, I., Johnson, A.E., et al.: An open access database for the evaluation of heart sound algorithms. Physiol. Meas. **37**(12), 2181 (2016)
9. Oppenheim, A.V., Verghese, G.C.: Signals, Systems and Inference. Pearson, Boston (2015)
10. Ozenne, B., Subtil, F., Maucort-Boulch, D.: The precision-recall curve overcame the optimism of the receiver operating characteristic curve in rare diseases. J. Clin. Epidemiol. **68**(8), 855–859 (2015)

11. Potes, C., Parvaneh, S., Rahman, A., Conroy, B.: Ensemble of feature-based and deep learning-based classifiers for detection of abnormal heart sounds. In: Computing in Cardiology Conference (CinC), pp. 621–624. IEEE (2016)
12. Rubin, J., Abreu, R., Ganguli, A., Nelaturi, S., Matei, I., Sricharan, K.: Classifying heart sound recordings using deep convolutional neural networks and mel-frequency cepstral coefficients. In: Computing in Cardiology Conference (CinC), pp. 813–816. IEEE (2016)
13. Saito, T., Rehmsmeier, M.: The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. PLoS ONE **10**(3), e0118432 (2015)
14. Springer, D.B., Tarassenko, L., Clifford, G.D.: Logistic regression-hsmm-based heart sound segmentation. IEEE Trans. Biomed. Eng. **63**(4), 822–832 (2016)
15. Wallace, B.C., Small, K., Brodley, C.E., Trikalinos, T.A.: Class imbalance, redux. In: 2011 IEEE 11th International Conference on Data Mining (ICDM), pp. 754–763. IEEE (2011)
16. Whitaker, B.M., Suresha, P.B., Liu, C., Clifford, G., Anderson, D.: Combining sparse coding and time-domain features for heart sound classification. Physiol. Meas. **38**, 1701–1713 (2017)
17. Zabihi, M., Rad, A.B., Kiranyaz, S., Gabbouj, M., Katsaggelos, A.K.: Heart sound anomaly and quality detection using ensemble of neural networks without segmentation. In: Computing in Cardiology Conference (CinC), pp. 613–616. IEEE (2016)

Computational Modeling

Analysis of the Co-purchase Network of Products to Predict Amazon Sales-Rank

Utpal Prasad(✉), Nikky Kumari, Niloy Ganguly, and Animesh Mukherjee

Indian Institute of Technology Kharagpur, Kharagpur, India
utpaldps@gmail.com

Abstract. Amazon sales-rank gives a relative estimate of a product item's popularity among other items in the same category. An early prediction of the Amazon sales-rank of a product would imply an early guess of its sales-popularity relative to the other products on Amazon, which is one of the largest e-commerce hub across the globe. Traditional methods suggest use of product review related features, e.g., volume of reviews, text content of the reviews etc. for the purpose of prediction. In contrast, we propose in this paper for the first time a *network-assisted* approach to construct suitable features for prediction. In particular, we build a *co-purchase* network treating the individual products as nodes, with edges in between if two products are bought with one another. The way a product is positioned in this network (e.g., its centrality, clustering coefficient etc.) turns out to be a strong indicator of its sales-rank. This network-assisted approach has two distinct advantages over the traditional baseline method based on review analysis – (i) it works even if the product has no reviews (relevant especially in the early stages of the product launch) and (ii) it is notably more discriminative in classifying a popular (i.e., low sales-rank) product from an unpopular (i.e., high sales-rank) one. Based on this observation, we build a supervised model to early classify a popular product from an unpopular one. We report our results on two different product categories (CDs and cell phones) and obtain remarkably better classification accuracy compared to the baseline scheme. When the top 100 (700) products based on sales-rank are labelled as popular and the bottom 100 (700) are labelled as unpopular, the classification accuracy of our method is 89.85% (82.1%) for CDs and 84.11% (84.8%) for cell phones compared to 46.37% (68.75%) and 83.17% (71.95%) respectively from the baseline method.

1 Introduction

Revenue forecasting and sales prediction have recently become very active areas of research especially in the context of box office revenues of newly released movies [1–4]. Further, in almost all of these studies, analysis of the online reviews has been shown to be very effective in such forecasting/prediction. With e-commerce platforms becoming increasingly more popular it is very important for these businesses to be able to understand and, in fact, to early identify the

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 197–214, 2017.
https://doi.org/10.1007/978-3-319-72413-3_13

sales impact of their different products. Amazon, for instance, maintains sales-rank of every product item which is a number with 1 to 8 digits and captures the product's relative popularity and visibility in comparison to other products in the corresponding sub-category of products. Authors, publishers, marketplace sellers, and many other people and businesses use sales-rank data as an indicator of how well their products are selling. They also analyse sales-ranks to further predict how well a product may sell and to decide whether or not to at all sell a particular product on Amazon.

While the Amazon sales-rank has been the source of much speculation by publishers, manufacturers and marketers, Amazon does not itself release the details of its sales-rank calculation algorithm. Further, it has been observed that sales-rank measures a product's popularity only in its corresponding sub-category. It does not bear any direct correlation with the absolute sales of the product; in fact, it has been conjectured that the rate of growth of the sales-rank of a product is high if the product has almost no sales history while it is very slow if the product has a long sales history[1]. An early prediction of sales-rank can enable the producers and consumers to predict the overall future acceptance of a product item in comparison to other items in the e-market. Further, it could immensely help in estimating the product's exposure on Amazon in future and enable the design of suitable early intervention mechanisms geared toward promoting the product.

In this work, we present an elegant approach to automatically distinguish early on time the popular (i.e., low sales-rank) products from the unpopular (i.e., high sales-rank) ones. Unlike traditional approaches that suggest analysis of reviews (e.g., volume of reviews, latent sentiment in reviews, interval between consecutive reviews) we propose a *network-assisted* scheme for the classification of the popular products.

Key contributions:

- We define a *co-purchase* network of products belonging to the same category where each node is a product and two nodes are connected by an edge if they are bought together. For our experiments, we construct networks for two different product categories – CDs and cell phones. In the first network all the nodes are CDs and two nodes are connected if one CD is co-purchased with another. Similarly, in the second network, each node is a cell-phone and two nodes are connected by an edge if the corresponding cell phones are co-purchased[2].
- We quantify how a product is positioned in the co-purchase network by extracting various structural properties like clustering co-efficient, betweenness, Pagerank, eigenvector and closeness centrality and community membership.

[1] See discussions on sales-rank calculation at https://kdp.amazon.com/community/message.jspa?messageID=562491.

[2] Note that this construction is much different and certainly more non-trivial than a general co-purchase network of all products in which breads might also get linked to bleaches by virtue of being bought together sometimes from the store.

- Since there is no suitable baseline available for this problem, as an additional objective, we define a very competitive baseline built on features extracted through extensive analysis of the reviews. To prepare the baseline, we consider the number of times a product is co-purchased with other products. In addition, we extract various linguistic features like the extent of anger, sadness, negative emotion in the user reviews per product as well as certain general features like the volume of reviews, percentage of fake reviews, dwell time and entropy of ratings.
- We build a binary classifier based on the network features to early classify a popular product from an unpopular one. We compare the performance of the classifier with that built from the baseline features based on online reviews.

Notable Observations:

- Our proposed method can work even in the absence of any reviews for a product. This is especially important at the early stages of a product launch when online reviews for a product are scarce.
- The network features that we propose are notably more discriminative than the baseline features based on review analysis.
- The classifier that we build, in particular, for two different product categories (CDs and cell phones) results in a remarkably better classification accuracy compared to the baseline scheme. In case we label the top 100 (700) products based on sales-rank as popular and the bottom 100 (700) as unpopular, the classification accuracy of our method is 89.85% (82.1%) for CDs and 84.11% (84.8%) for cell phones compared to 46.37% (68.75%) and 83.17% (71.95%) respectively from the baseline method.

The rest of the paper is organised as follows. In Sect. 2 we discuss the relevant past literature. We describe our dataset in detail in Sect. 3. We next define the co-purchase network and study the metrics describing the position of a node in the network that can discriminate the popular products from the unpopular ones in Sect. 4. In Sect. 5 we describe the baseline features. We present the classification results in Sect. 6. Finally, we conclude in Sect. 7 by summarizing our key contributions and outlining some future directions.

2 Related Work

Revenue forecasting and sales prediction has been extensively studied for the entertainment industry especially in the context of box office revenues for motion pictures [1–4]. Almost all of these studies focus on mining online reviews to predict the sales performance. For instance, in [1,3] the authors use online reviews to construct novel diffusion models that are capable of accurate revenue forecasting. In [2], the authors analyse online ratings, and in particular, identify the valence of user ratings as a good predictor for motion picture revenue. Further, in [4], the authors perform detailed sentiment analysis of the review text to predict the sales.

With an exponential increase in the use of e-commerce platforms, there is an increased importance for the study of sales impact of different products on these platforms. While there has been a huge volume of work to design accurate recommender systems for e-commerce platforms to help consumers choose the best products [5–7], very little attention has been paid to analyse the future impact of a product. A few works that are remotely associated to this task are as follows. [8] studies the on-line shopping behaviour to improve user engagement on e-commerce sites. Amazon data has been used to study image-based recommendations [9] and to build complimentary product networks [10]. Work has been done to analyze Amazon e-commerce reviews to understand the (i) helpfulness of the reviews [11] and the (ii) latent sentiment in the reviews [12]. [13] posits that identity-descriptive information in the reviews can improve product sales.

In contrast, to the above approaches, we present for the first time a network-assisted method based on the co-purchasing behavior of the consumers to distinguish early the popular products from the unpopular ones. Our method is unique as it can work even in the complete absence of the online reviews which is usually the case immediately after the product launch.

3 Dataset Description

We have used the Amazon product data shared by McAuley et al. [9,10] for our experiments. The dataset contains reviews and metadata pertaining to the different products on Amazon. It consists of nearly 140 million reviews spanning from May 1996 to July 2014. It includes various review related information like ratings, text, helpfulness votes as well as product metadata which includes descriptions, category information, price, brand, image features, also bought products, also viewed products, etc. There are multiple broad categories in the dataset like books, electronics, CDs, cells, clothings, etc. We consider two of the largest categories – cells and CDs – for our study. There are 3,749,004 reviews and 492,799 product metadata entries corresponding to the category of CDs. Similarly, there are 3,447,249 reviews and 346,793 product metadata entries corresponding to the category of cells. We consider only those products which have greater than 1 review per month on average from the period 2010 to 2013. There are 2,624 such CDs and 11,564 such cells. We sort the products in each category according to sales-rank and consider various windows for the purpose of classification. We take top (bottom) k products with $k = 100$, 300, 500 and 700 from the sorted list of products and call them popular (unpopular). For each of these products we extract both network-centric as well as baseline features and build the binary classifier. As we shall show later, even for a weak separation at $k = 700$, the network-centric features perform notably better classification than the baseline features. Note that in all the classification experiments, we divide the dataset into training and test examples. All the products launched in the market before the start of 2013 (and are part of the list of the top (bottom) k products) are used for training while the rest are used for testing. We compute the features for each product in the training set using data till the end of 2012 and predict the sales-rank value of the newly launched products (after 2012) at the end of July 2014. This approach strictly ensures that there is no information leakage in our prediction scheme.

4 Network-Assisted Sales-Rank Characterization

In this section, we propose a novel network-assisted approach to characterize the sales-rank of Amazon products. In the rest of this section we define the network, extract important structural properties from this network indicating how a particular node is positioned and use these to establish strong differences between the popular and the unpopular products.

4.1 Co-purchase Network

We construct a co-purchase network of the products belonging to a category using the "also-bought" information available in the metadata. The co-purchase network has products as nodes and an edge between the nodes if two products are bought with one another. We remove all nodes from the network which have very low degree (≤ 1).

4.2 Structural Properties of the Network

In the rest of this section we analyse the different structural properties of the co-purchase network to identify how a particular node (product) is positioned in this network. We further show how this positional information distinguishes a popular product from an unpopular one. In all the results that we present, the popular class comprises top 300 products as per sales-rank and the unpopular class comprises the bottom 300 products as per sales-rank. Note that for all the network features, these results remain very similar even when one considers top 100, 500 or 700 products as per sales-rank in the popular class and respectively bottom 100, 500 and 700 products in the unpopular class.

Clustering Coefficient. In our experiments, we use the definition of clustering coefficient based on triplets of nodes. For unweighted graphs, the clustering of a node u is the fraction of possible triangles that the node u is part of and is given by

$$c_u = \frac{2T(u)}{deg(u)(deg(u) - 1)}$$

where $T(u)$ is the number of triangles that the node u is part of and $deg(u)$ is the degree of u. A high clustering co-efficient indicates a densely connected neighborhood for a node. An interesting observation is that nodes corresponding to popular products tend to have higher clustering co-efficient, i.e., a denser neighborhood than the unpopular products in the co-purchase network (see Figs. 1 and 2 for the two product categories). This possibly indicates that popular products tend to be co-purchased with other popular products thus forming dense neighborhoods or "rich-clubs" [14] of popular products.

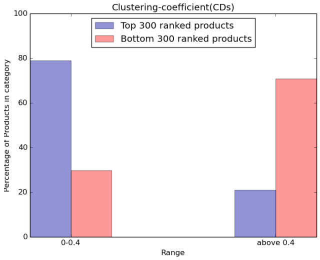


Fig. 1. Percentage of products from the two class vs clustering coefficient buckets (for CDs).

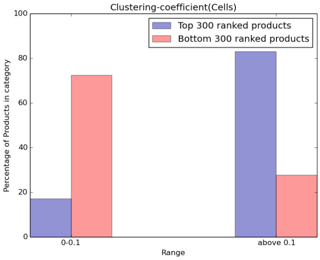


Fig. 2. Percentage of products from the two class vs clustering coefficient buckets (for cells).

Betweenness. Betweenness of a node in a graph measures the extent to which the node lies on paths between other vertices. It is equal to the fraction of the number of shortest paths from all vertices to all others that pass through that node. Betweenness is given by the equation:

$$g(v) = \frac{\sum_{s!=v!=t} \sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is the number of shortest paths between the vertex pair (s, t) and $\sigma_{st}(v)$ is the number of those paths among these that pass through v.

Betweenness of a node in the co-purchase network is a reflection of how often the product corresponding to this node bridges two or more unrelated products. Since popular products are often bought with many other products, they tend to have higher betweenness. This is apparent from Figs. 3 and 4 where we plot for both the product categories, the percentage of popular and unpopular products in low and high buckets of betweenness. Most of the unpopular products have low betweenness while the popular ones have high betweenness.

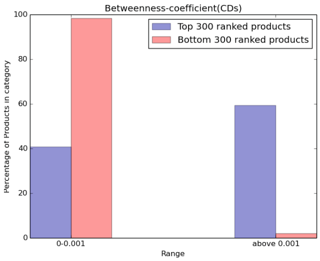


Fig. 3. Percentage of products from the two classes vs betweenness buckets (for CDs).

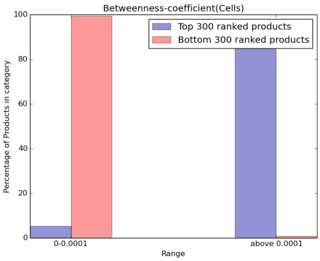


Fig. 4. Percentage of products from the two class vs betweenness buckets (for cells).

Closeness. Closeness centrality of a node u is the reciprocal of the sum of the shortest path distances from u to all $n - 1$ other nodes (assuming there are n nodes in the network). Since the sum of the distances depends on the number of nodes in the graph, closeness is normalized by the sum of the minimum possible distances $n-1$. If the graph is not completely connected, we compute the closeness centrality for a node corresponding to its own connected component. Mathematically, the closeness is:

$$C(u) = \frac{n-1}{\sum_{v=1}^{n-1} d(v,u)}$$

where $d(v, u)$ is the shortest path distance between v and u.

In the co-purchase network, a popular product will be also bought with many other products and should therefore be close to most of the other nodes in the network. This is apparent from Figs. 5 and 6 where we plot for the two product categories, the percentage of popular and unpopular products in low and high buckets of closeness. For both the categories, the lower bucket of closeness has a large percentage of unpopular products while the higher bucket has a large percentage of popular products.

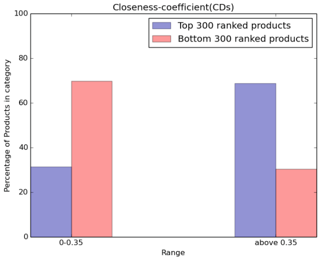


Fig. 5. Percentage of products from the two classes vs closeness buckets (for CDs).

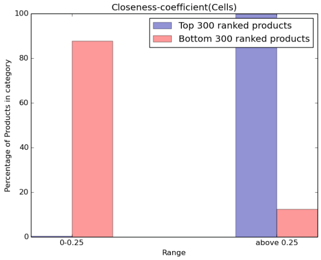


Fig. 6. Percentage of products from the two class vs closeness buckets (for cells).

Eigenvector. Eigenvector centrality is a measure of the recursive influence of a node in a network. It assigns relative scores to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. Popular products tend to have a higher eigenvector centrality (see Figs. 7 and 8).

Pagerank. Pagerank works by counting the number and quality of links to a node (product) to determine a rough estimate of how important the product is. The underlying assumption is that more important products are likely to receive more links from other websites. Popular products are likely to have better Pagerank centrality (see Figs. 9 and 10).

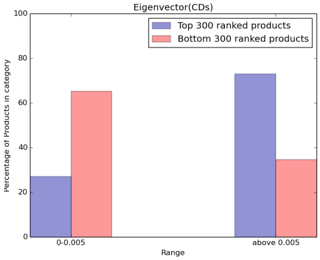


Fig. 7. Percentage of products from the two classes vs eigenvector buckets (for CDs).

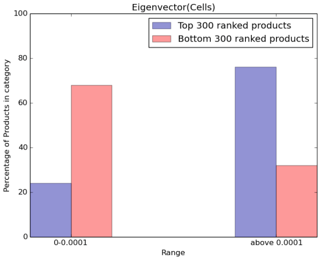


Fig. 8. Percentage of products from the two classes vs eigenvector buckets (for cells).

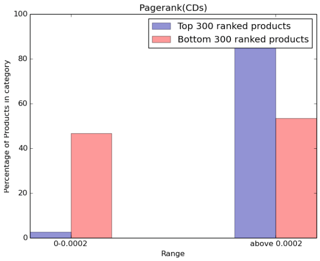


Fig. 9. Percentage of products from the two classes vs Pagerank buckets (for CDs).

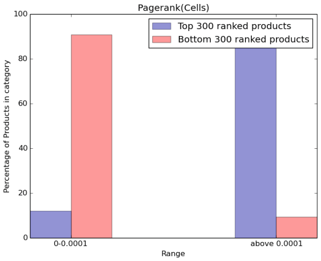


Fig. 10. Percentage of products from the two classes vs Pagerank buckets (for cells).

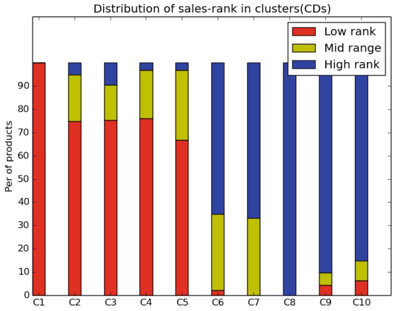


Fig. 11. Stacked bar plots showing the percentage of products from the different sales-rank classes in some representative communities (CDs).

Community Membership. We use the popular Louvain [15] community detection algorithm to find the community structure of the co-purchase network. We observe that the community memberships of the popular products are very distinct from that of the unpopular ones. Precisely, in a majority of

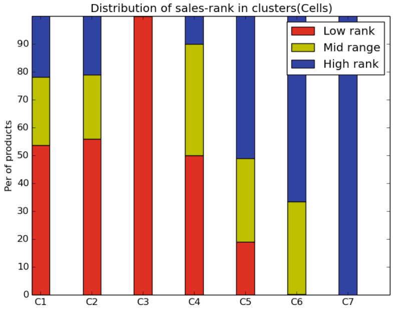


Fig. 12. Stacked bar plots showing the percentage of products from the different sales-rank classes in some representative communities (cells).

cases the communities are either mostly composed of only the higher sales-rank (i.e., the bottom class) products or mostly composed of only the lower sales-rank (i.e., the top class) products (see Figs. 11 and 12 for the two product categories respectively).

5 Baseline Features Based on Review Analysis

Since there is no baseline available in the literature for this problem, as an additional objective, we design a set of competitive baseline features through an extensive analysis of the reviews. The idea of using the reviews for building a baseline is inspired by similar approaches used for box office revenue forecasting [1–4]. The set of features can be further classified into (i) general and (ii) linguistic features. We discuss each of these in the following two subsections.

5.1 General Features

In this subsection, we shall define some general features extracted from the pattern of online reviews of the products. In all the results that we present, the popular class comprises the top 300 products as per sales-rank and the unpopular class comprises the bottom 300 products as per sales-rank. The separation weakens as more products from the top and the bottom zones of the sales-rank are included into the respective classes.

Volume of Reviews. Volume of reviews is expressed as the total number of reviews for a product in the span 2010–2013. Analysis of this feature for both the classes of products shows that popular products have a higher number of reviews as compared to unpopular products (see Figs. 13 and 14 for the two product categories).

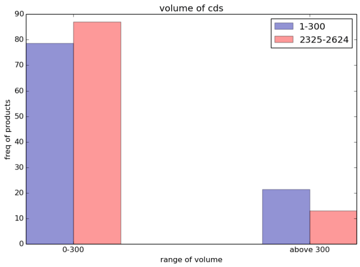


Fig. 13. Percentage of products from each class vs the volume of reviews (for CDs).

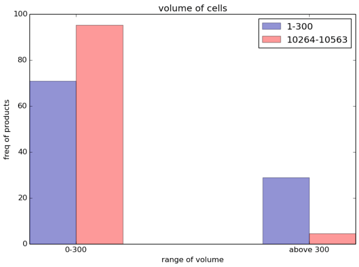


Fig. 14. Percentage of products from each class vs the volume of reviews (for cells).

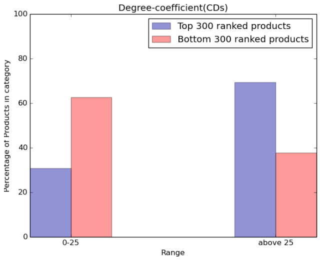


Fig. 15. Percentage of products from the two class vs co-purchase count buckets (for CDs).

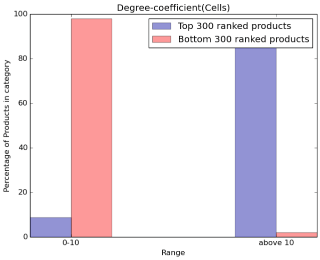


Fig. 16. Percentage of products from the two class vs co-purchase count buckets (for cells).

Number of Co-purchases. This feature counts the number of products with which a particular product is co-purchased. Note that this is also the degree of a node in the co-purchase network and we treat the same as a baseline feature to specifically show later that this trivial measure is not as good an indicator of sales-rank as the other non-trivial network measures described in the previous section. Figures 15 and 16 shows for the two product categories that a larger percentage of popular products fall in the high co-purchase count bucket while a larger percentage of unpopular products fall in the low co-purchase count bucket.

Percentage of Fake Reviews. We calculate the percentage of fake reviews by first classifying the reviews as real or fake using a Naïve-Bayes supervised learning approach based on standard tf-idf features. For training the classifier, we have used a proxy dataset available from Yelp that has a huge collection of review text which are already marked fake [16]. In Figs. 17 and 18 we observe that the percentage of fake reviews is more for an unpopular product as compared to a popular product. A possible reason for this is that certain users (for instance, the sellers themselves) might have vested interest in promoting an unpopular product.

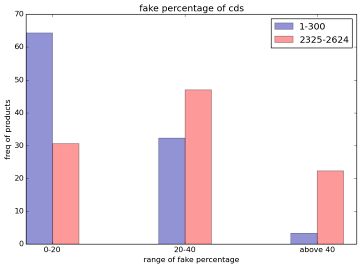


Fig. 17. Percentage of products from each class vs volume of fake reviews (for CDs).

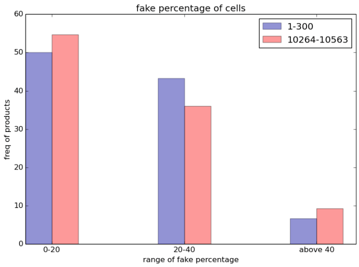


Fig. 18. Percentage of products from each class vs volume of fake reviews (for cells).

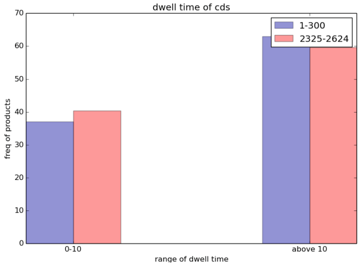


Fig. 19. Percentage of products from each class vs dwell time (for CDs).

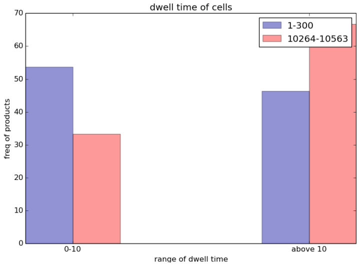


Fig. 20. Percentage of products from each class vs dwell time (for cells).

Dwell Time. In the context of the current problem, we define dwell time as the continuous stretch in number of months for which a product is receiving reviews. We observe from our analysis that average dwell time is more for popular products (see Figs. 19 and 20 for the two product categories).

Entropy of Ratings. Entropy of ratings is calculated as $-\sum_1^5 p_i \log p_i$, where p_i denotes the probability of a rating i across all the reviews of a product. A high entropy would indicate that the product receives diverse ratings from users while a low entropy would indicate similar ratings from all users. We see that unpopular products have higher entropy and thus users have more diverse/mixed opinion about them (see Figs. 21 and 22 for the two product categories).

5.2 Linguistic Features

In this section we perform extensive analysis of the review text to design various features based on the language structure. One again, the popular class comprises the top 300 products as per sales-rank and the unpopular class comprises the bottom 300 products as per sales-rank. The separation weakens as more products from the top and the bottom zones of the sales-rank are included into the respective classes.

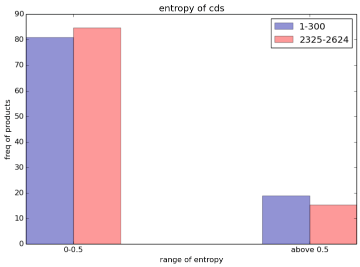


Fig. 21. Percentage of products from each class vs entropy of ratings (for CDs).

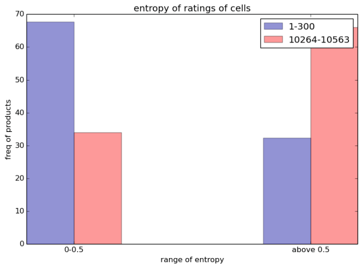


Fig. 22. Percentage of products from each class vs entropy of ratings (for cells).

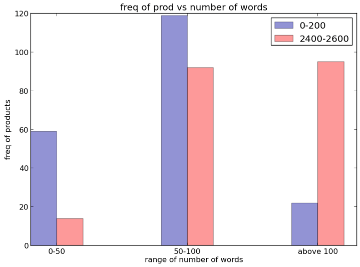


Fig. 23. Number of products from each class vs number of words (for CDs).

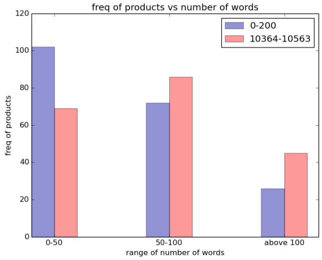


Fig. 24. Number of products from each class vs number of words (for cells).

Number of Words. We have taken the number of words in a review as one of the initial linguistic features. We observe that unpopular products have more lengthy reviews (see Figs. 23 and 24) and this might be due to the fact that people are usually not very satisfied with the product which compel them to give lengthy comments so that the product can be improved.

Word Diversity. Word diversity is defined as the entropy of fraction of words in a particular review which is calculated using the formula $-\sum_1^N p_i \log p_i$ where p_i denotes the count of a particular word i in a review divided by total length of the review i.e., N. We see that for popular products, entropy is less (see Figs. 25 and 26) as compared to the unpopular products which signify that reviews for popular products are more well-formed and linguistically better structured.

Anger. We next investigate some of the interesting linguistic factors using the LIWC[3] (Linguistic Inquiry and Word Count) text analysis tool [17]. The tool provides, as output, percentage of words in different categories for an input text. The categories are broadly divided into linguistic (21 dimensions like pronouns,

[3] http://liwc.wpengine.com/.

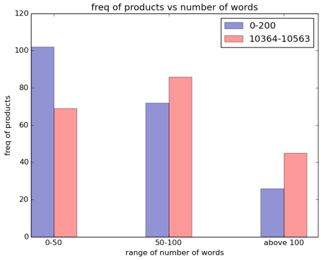


Fig. 25. Number of products from each class vs diversity of words (for CDs).

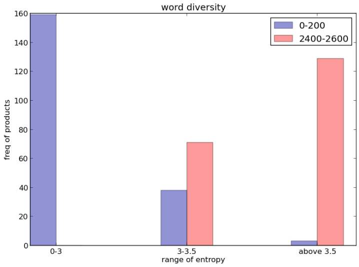


Fig. 26. Number of products from each class vs diversity of words (for cells).

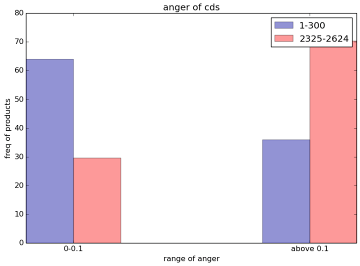


Fig. 27. Percentage of products from each class vs the extent of LIWC anger feature (for CDs).

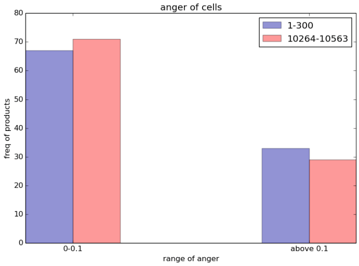


Fig. 28. Percentage of products from each class vs the extent of LIWC anger feature (for cells).

articles etc.), psychological (41 dimensions like affect, cognition etc.), personal concern (6 dimensions), informal language markers and punctuation apart from some general features like word count, words per sentence etc. The first factor that we find well differentiates between popular and unpopular products is the extent of anger in the reviews. While popular products have low anger content in their reviews, the unpopular ones have high anger content (see Figs. 27 and 28 for the two product categories.).

Sad. Next we report the extent of sadness in the review text from the suite of LIWC features. Figures 29 and 30 show that products with low average sadness values in their reviews are more probable to belong to the popular class while the opposite is true for the unpopular class.

Negative Emotion. Another discriminating LIWC feature is the extent of negative emotions present in the review text. Figures 31 and 32 show that a product with low average negative emotion value is more probable to belong to the popular class of products. The opposite is true for the unpopular class.

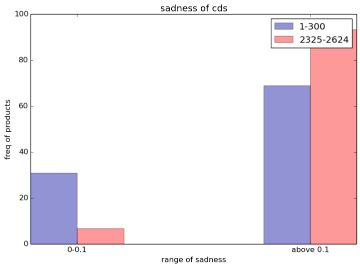


Fig. 29. Percentage of products from each class vs the extent of LIWC sadness feature (for CDs).

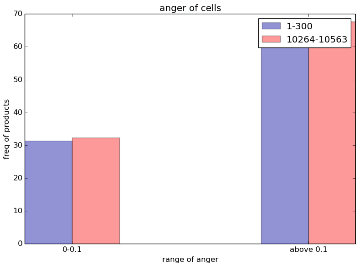


Fig. 30. Percentage of products from each class vs the extent of LIWC sadness feature (for cells).

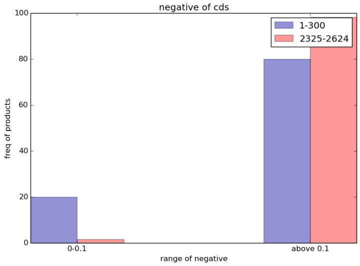


Fig. 31. Percentage of products from each class vs the extent of LIWC negative emotion feature (for CDs).

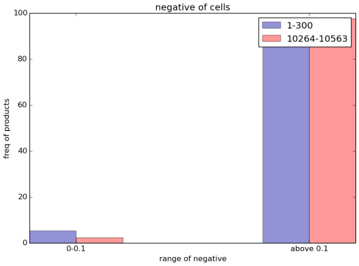


Fig. 32. Percentage of products from each class vs the extent of LIWC negative emotion feature (for cells).

All the results in this section together clearly portray that none of the review based features are as distinguishing as the network-centric features proposed in the previous section.

6 Classification

In this section, we design a binary classifier (linear SVM) to early predict the class of a product using the network features described earlier. We assign positive and negative labels (popular/unpopular) against the products by choosing appropriate thresholds for defining popularity. The top (bottom) k $(100, 300, 500, 700)$ products are classified as popular (unpopular). We split the products of a category into training and test sets. All the products launched before the start of 2013 are used for training while the rest are used for testing. We compute the features for each product in the training set using data till the end of 2012 and predict the sales-rank value of the newly launched products (i.e., since 2013 beginning) at the end of July 2014. This ensures a fair distribution of training set across the two classes. We present our results for two different product categories and different values of $k = 100, 300, 500, 700$ (see Table 1).

Table 1. Accuracy of classification (baseline and our method) for the two product categories.

Accuracy (CDs)		
Number of products	% Accuracy (baseline features)	% Accuracy (network features)
100	46.37	**89.85**
300	61.58	**92.68**
500	69.23	**86.15**
700	68.75	**82.10**
Accuracy (cells)		
Number of products	% Accuracy (baseline features)	% Accuracy (network features)
100	83.17	**84.11**
300	71.95	**84.15**
500	68.56	**81.44**
700	71.95	**84.8**

In parallel, we also learn an SVM classifier using the baseline features outlined above. Once again, we produce results for both the product categories as well as different values of k. In all cases, our network assisted approach overwhelmingly outperforms the baseline scheme.

Note that the classification based on network features by far outperforms the baseline features. Even for a weak separation of $k = 700$ we obtain an accuracy improvement of $\sim$19% for CDs and $\sim$17.8% for cells. Note that this result also shows that the trivial feature of the count of co-purchases (also the degree of a node in the co-purchase network) used as a part of the baseline features is not as effective in predicting the popularity class as the more non-trivial features based on network properties.

Importance of the network features: We further perform a χ^2 test to identify the importance of the individual network features in the classification for both the product categories. We find (see Table 2) that **community membership** is a very discriminative feature for both the products, **closeness** is more discriminative for cells while **clustering coefficient** is more discriminative for CDs.

Network + baseline features: A final question that one might ask is whether, the performance of the classifier improves if the baseline features are available and are used in addition to the network features. To answer this question, we report in Table 3 the accuracy we obtain by using both the network and the baseline features. As one would expect, in all the case, the improvements resulting from this combination is only marginal.

Table 2. χ^2 ranking of the network features.

χ^2 values		
Feature	% CDs	% Cells
Community membership	437.62	6.55
Eigenvector	3.31	0.27
Pagerank	0.1	0.01
Closeness	0.65	35.77
Clustering coefficient	14.40	1.40
Betweenness	0.94	0.91

Table 3. Accuracy of classification (network + baseline features) for the two product categories.

Accuracy (CDs)	
Number of products	% Accuracy (network + baseline features)
100	84.05
300	92.64
500	90.08
700	86.93
Accuracy (cells)	
Number of products	% Accuracy (network + baseline features)
100	83.17
300	84.48
500	84.06
700	84.62

7 Conclusion

In this paper, we presented a network-assisted method to early predict the popularity class of a product on Amazon. In particular we made the following contributions:

- We defined a co-purchase network and computed various positional information about individual nodes; these positional information turn out to be strong indicators of popularity of a product.
- Since there was no standard baseline for this work, we proposed a baseline contrived from co-purchase count, reviews and ratings feature of a product.
- We devised a classifier based on the network properties and showed that it outperforms the baseline by large performance margins. In specific, even for a weak separation between the popular and the unpopular products the improvement in accuracy is quite high.

– Among the network features, community membership, closeness and clustering coefficient metrics are found to be quite discriminative.

Such an early prediction could be extremely helpful for the entire business including the Amazon group, the sellers, investors and marketers portraying a clear picture of the sales impact of a product. This would also facilitate the design of suitable intervention measures to promote certain products to enhance the eventual sales figure as well as to decide if some product should be withdrawn from the marketplace. Through rigorous experiments we show that our results remarkably outperform the baseline approach built from traditional review analysis.

In future, we wish to perform similar analysis for other similar e-commerce platforms and identify if the network-assisted method has universal implications. Further, we would also like to investigate if such network-centric methods could be leveraged to study other market characteristics such as purchasing behavior of the customers, selling behavior of the sellers, the speed of sales etc.

References

1. Dellarocas, C., Awad, N., Zhang, X.M.: Using online reviews as a proxy of word-of-mouth for motion picture revenue forecasting. SSRN Electron. J. (2004)
2. Dellarocas, C., Awad, N., Zhang, X.M.: Using online ratings as a proxy of word-of-mouth in motion picture revenue forecasting. Working Paper (2005)
3. Dellarocas, C., Zhang, X.M., Awad, N.: Exploring the value of online product reviews in forecasting sales: the case of motion pictures. J. Interact. Mark. **21**, 23–45 (2007)
4. Yu, X., Liu, Y., Huang, J.X., An, A.: Mining online reviews for predicting sales performance: a case study in the movie domain. IEEE TKDE **24**, 720–734 (2012)
5. Schafer, J.B., Konstan, J., Riedl, J.: Recommender systems in e-commerce. In: Proceedings of the 1st ACM Conference on Electronic Commerce, pp. 158–166 (1999)
6. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Analysis of recommendation algorithms for e-commerce. In: Proceedings of the 2nd ACM Conference on Electronic Commerce, pp. 158–167 (2000)
7. Wei, K., Huang, J., Fu, S.: A survey of e-commerce recommender systems. In: International Conference on Service Systems and Service Management, pp. 1–5 (2007)
8. Sharma, N.V., Khattri, V.: Study of online shopping behavior and its impact on online deal websites. Asian J. Manag. Res. **3**(2), 394–405 (2013)
9. McAuley, J., Targett, C., Shi, Q., van den Hengel, A.: Image-based recommendations on styles and substitutes. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 43–52. ACM (2015)
10. McAuley, J., Pandey, R., Leskovec, J.: Inferring networks of substitutable and complementary products. In: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM (2015)
11. Mudambi, S.M., Schuff, D.: What makes a helpful review? A study of customer reviews on amazon.com. MIS Q. **34**(1), 185–200 (2010)

12. Jo, Y., Oh, A.H.: Aspect and sentiment unification model for online review analysis. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 815–824. ACM (2011)
13. Forman, C., Ghose, A., Wiesenfeld, B.: Examining the relationship between reviews and sales: the role of reviewer identity disclosure in electronic markets. Inf. Syst. Res. **19**(3), 291–313 (2008)
14. Colizza, V., Flammini, A., Serrano, M.A., Vespignani, A.: Detecting rich-club ordering in complex networks. Nat. Phys. **2**, 110–115 (2006)
15. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech: Theory Exp. **10**, P1000 (2008)
16. Mukherjee, A., Venkataraman, V., Liu, B., Glance, N.: What yelp fake review filter might be doing? In: Proceedings of the 7th International AAAI Conference on Weblogs and Social Media, pp. 409–418. ACM (2013)
17. Pennebaker, J.W., Boyd, R.L., Jordan, K., Blackburn, K.: The development and psychometric properties of liwc2015. UT Faculty/Researcher Works (2015)

Model-Free Optimal Control: A Critical Analysis

Vijaysekhar Chellaboina(✉)

Mahindra École Centrale, Hyderabad, India
vijay@mechyd.ac.in

Abstract. In this note, we present a critical analysis of machine learning techniques for applications involving optimal (feedback) control. Specifically, we will focus on the question of using reinforcement learning and other similar techniques in providing provably stable optimal controllers.

1 Introduction

The traditional control methodology is heavily based on the assumption that dynamical systems are represented in terms of clearly defined mathematical models [3,14]. The two corner stones of control theory are stability and performance. While performance is a standard requirement in any design process, stability is inherently connected with the dynamic nature of dynamical systems. The definitions of stability are intricately related to the type of model used. Such models are either derived from first principles or through a rigorous system identification techniques based on experimental data [11,20].

An ideal control methodology would be a model-free approach for deriving optimal controllers (which may be model-free themselves) just based on the input-output data of the system. At this time, machine learning techniques such as reinforcement learning come to close to providing a model-free approach for optimal control [24].

In this note, we provide a brief overview of both optimal control and machine learning techniques. In the discussion of optimal control, the need for the concept of the state and the state-space approach are considered in relation to stability of a system. In contrast, the overview of machine learning shows the difficulty in identifying either state or stability of systems consisting of machine learning *blocks*.

2 Optimal Feedback Control

In this section, we present a basic description of the problem of optimal control starting with a general description of dynamical systems, stability, performance, optimal, robust, and adaptive control. For more details on these well established topics one may refer to many of the standard references such as [2,3,6,14,22].

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 215–222, 2017.
https://doi.org/10.1007/978-3-319-72413-3_14

2.1 Dynamical Systems and Control

A *dynamical system* is a system that changes with time. Specifically, the complete description of a dynamical system consists of *input* and *output signals* (which are functions of time) and the relation between the input signals and output signals. Let $\boldsymbol{\mathcal{U}}$ denote the set of input signals $u : \mathbb{T} \to \mathbb{R}$ and $\boldsymbol{\mathcal{Y}}$ denote the set of output signals $y : \mathbb{T} \to \mathbb{R}$. Let the relation between u and y be given by a mapping $\boldsymbol{\mathcal{G}} : \boldsymbol{\mathcal{U}} \to \boldsymbol{\mathcal{Y}}$ so that $y = \boldsymbol{\mathcal{G}}u$, $u \in \boldsymbol{\mathcal{U}}$ so that the description of the dynamical system is complete by specifying $\boldsymbol{\mathcal{G}}$, $\boldsymbol{\mathcal{U}}$, and $\boldsymbol{\mathcal{Y}}$. If $\mathbb{T} = \mathbb{R}$ then the system is a *continuous-time* system and if $\mathbb{T}$ is a finite set then it is a *discrete-time* system. In this section, we focus only on the continuous-time version. Analogues for discrete-time version can be easily developed and are well documented in the literature.

In the case where $\boldsymbol{\mathcal{G}}$ is a linear mapping, the description may be given in terms of the Laplace transforms of the input and output signals, that is, $Y(s) = G(s)U(s)$, where $G(s)$ is called the *transfer function* of the system. A system described by a transfer function may be equivalently represented in its *state-space* form given in terms of ordinary differential equations

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{1}$$

$$y(t) = Cx(t) + Du(t), \tag{2}$$

where $x(t)$ is the *state* vector and A, B, C, D are *system matrices* are such that $G(s) = C(sI - A)^{-1}B + D$. Such a state-space description in terms of a (vector) ordinary differential equation is always possible if the transfer function is *real rational* and *proper*, that is, $G(s)$ is described in terms of a ratio of real polynomials with the numerator order less than or equal to that of the denominator. Even in the more general case, under mild technical assumptions, it is always possible to represent a linear system in a state-space form involving infinite-dimensional states. The most general description of a dynamical system is in terms of a state-function which maps initial state and inputs to the state at a future time [6,7,15].

In practice, the output signals are typically signals that can be measured using an instrument or a function of such signals. The input signals are typically divided into two categories (*i*) control inputs and (*ii*) disturbance inputs or noise. A *control problem* can then be stated as *determination of appropriate control input signals so that specific output signals follow a desirable pattern.* An *optimal control problem* is *determination of appropriate optimal control input signals so that specific output signals follow a desirable pattern and maximize a chosen performance.* It is understood that the performance will be in terms of only the input and the output signals.

2.2 Stability and State-Space Models

In control theory, every control system is designed for (*i*) stability and (*ii*) performance. The need for performance is clear and has been included in the control problem statement above. However, neither the definition nor the need for stability is obvious. The difficulty starts with the fact that there are numerous definitions of stability. The definitions of stability can be broadly classified into two categories (*i*) bounded-input, bounded-output (BIBO) stability and (*ii*) equilibrium-state stability.

A *dynamical system* is *BIBO stable* if for every bounded input, the output remains bounded. It should be noted that input and output signals are functions of time and hence there is no unique way of measuring the size (in terms of a norm) of these signals and hence the same system (represented by, say, $\mathcal{G}$) may be BIBO stable with respect to certain choice of input-output norm pairs but not stable with respect to other choices. The most standard choice (though not necessarily most natural) for these norms is the Euclidean norm or the L_2-norm. The *classical control theory* provides stability results for transfer functions has the interpretation of BIBO stability with the L_2-norm.

As described above, systems can also be represented in a state-space form. And for the state-space model, one may identify special states called equilibria. A state is called an *equilibrium* if the system (in the absence of inputs) starts in an equilibrium state then it remains there. An equilibrium is said to be *stable* if the system starts close to an equilibrium then it remains close to the equilibrium and approaches the state asymptotically. If the system has only one equilibrium and it is stable (as per the definition above) then such a system may be called as a *stable system*. The definition of equilibrium stability given here is known as *asymptotic stability* in the literature. There are many more equally interesting forms of stability (all connected to one another) but will not be discussed here.

The reader may be wondering if there is a relation between these two broad categories of stability. The connection between the two categories is most interesting. *A linear system is BIBO stable if and only if it is asymptotically stable.* The case of nonlinear systems is slightly more complicated. There are classical results proving that equilibrium-state stability implies BIBO stability. Indeed most of the results in the literature on BIBO stability of nonlinear systems rely on equilibrium-state stability.

It should be noted that the concept of equilibrium-state stability is inherently connected to the definition of a state. Though neither a state-space description nor the concept of equilibrium-state stability is necessary for defining stability of a dynamical system but it is interesting to note that most results on stability are given for state-space models. The above discussion clearly shows that state-space description and the corresponding stability notions makes it convenient for designing a control system. However, it is not clear if these notions are absolutely necessary.

The popularity of state-space approach has a strong underlying reason. Most physical systems can be modeled using first principles (such as Newton's laws of motion) and state-space descriptions are very natural. In such case, the state of the system is identified with physical variables. It is therefore important that the control systems for physical systems are designed such that the entire state remains bounded. Hence, nonlinear system control is almost synonymous with control for equilibrium-state stability and state-boundedness (see [14] and references therein). However, this is not the case if the system is not physical (for example, economic or socio-economic systems).

Are the concepts of state-boundedness and equilibrium-state stability essential for designing control systems?

2.3 Optimal Robust Control

In the previous section, we described the importance given to the concept of stability in systems theory. Here we focus on the issue of optimality and incomplete knowledge of system models. The most important goal in designing control systems is to design for best performance while satisfying all physical constraints. This is the subject of optimal control. Design of an optimal control for a given system requires the complete knowledge of the system. For example, in the linear case, the so called LQR or Kalman filters are optimal controllers (for specific performance criteria) and require the exact knowledge of the system matrices [2].

In practice, it is almost impossible to have the exact knowledge of system and hence every model has parameters or functions that are uncertain. Hence, the stability results will have to be extended for models where part of the model is unknown or uncertain. Such stability results form the concept of *robust stability* [8,14,27]. For example, if a system satisfies an input-output property known as *passivity* [14] and there is no other information available about the system one can design a passive controller to make the overall system stable. More generally, the concept of dissipativity can be used developed stability results for systems with different classes of model uncertainty [14]. The dissipativity-based results are applicable for systems with and without an explicit state-space characterization. These provide methods for designing a stabilizing controller for a set of (uncertain) systems and the performance obtained will be the worst-case performance. Hence, the optimal control methodology based on robust stability concepts can only be used to design controllers maximizing the worst-case performance. If the uncertainty set if *large* then the optimal performance will be poor.

2.4 Adaptive Control

An alternate method to control uncertain systems is to use the idea of *adaptive control* [4,17,22]. The main idea of adaptive control is to adapt the parameters of the controllers so that the performance of the closed-loop system is optimal at each and every operating point. In the 1950's and the 1960's, NASA had an extensive research-airplane program to test the adaptive control methodology and it was finally shut down due to a fatal accident [18]. The analysis of the accident revealed that the failure is due to overall stability issues (as opposed to stability at every operating point). This led to the development of stable or provably correct adaptive systems [22] based on rigorous Lyapunov approach applied to state-space models with parametric uncertainty. Stable adaptive control, as compared to robust control, provides a framework for stabilization as well as optimal performance. See [10] for a recent analysis of the NASA X-15 program and the lessons learned from it.

Stability takes priority over performance

3 Machine Learning for Control

In this section, we discuss the relevance of machine learning techniques in the context of feedback control with specific focus on the issues of stability, performance, and uncertainty. First, we present a brief overview of machine learning techniques and their specific role in control systems. For an introduction to machine learning techniques and its allied topics see [1,5,13,21,24,26] and numerous references within.

3.1 Overview of Machine Learning

The ever increasing ability to manipulate and to compute with large sets of data makes it possible to implement many of the machine learning techniques for a variety of applications. *Machine learning* may simply be defined as extracting *information* from *data* using computing machines. Extracting information from data is as old as science and every fundamental laws of nature is an example of such extraction. Hence, every model discovery is such an example. If a computational tool is utilized in extracting such information or a model from data then the methodology is dubbed as *artificial intelligence* or more modestly *machine learning*.

Machine learning techniques process large data (of input and output of a given system) to essentially provide a *black-box* model of the system, which may then be used in predicting output for an input that is not part of the original data. The black box may contain any number of models available including *neural networks*, *support vector machines*, *decision trees*, and a variety of other underlying models. These models are derived using error minimization techniques including *back propagation algorithms* and *reinforcement learning algorithms*. Since these techniques are primarily based on the paradigm of processing large amount of data the models thus obtained may also be significantly high dimensional, essentially rendering them useless in terms of rigorous mathematical analysis. Hence, the abstraction of data into one of the machine learning models is dubbed as *model-free* approach. Here, the model-free approach also refers to the fact that a model is not developed from the first principles (laws of nature) but only from the available data (input as well as output).

Machine learning provides a model-free data-based approach to predict outputs for unknown inputs

3.2 Model-Free Control

The literature on machine learning applications to control systems is a lot sparser compared to that of other applications. See, for example, [9,19,23,25]. With exception of neural network based control (see for example, [12,16]) most of the machine learning based control do not focus on the proof of stability. The primary focus of such literature is on performance maximization or optimizing parameters of a stabilizing controller. The available literature on machine learning applications to control systems can be broadly classified as follows:

(*i*) Neural-network or similar model-based optimal control where the weights/ parameters of the model are adapted for stabilization and optimal performance. In this case, the models are invariably state-space based and the stability proof is in terms of equilibrium-state using Lyapunov-like approaches. This is simply a large scale version of model-based state-space approach to control.

(*ii*) Reinforcement-learning or a similar technique is used to develop optimal controllers based only on (large amount of) input-output data with appropriate control problem stated in terms of input and output signals. In most of these cases, (BIBO) stability is an inherent quality of the system or no proof of stability is considered. This is truly a model-free approach to control, that is, no model is derived from first principles or the available large-scale model is not useful for formal analysis.

Note that the above model-free approach is ideal for systems which are inherently bounded (for example, finite-state machines). In this case, the reinforcement learning and other techniques can be used to extract maximum performance from the system. However, in the case of systems where the boundedness is not inherent to the system then it is not clear that the model-free approach is sound. Though there have been multiple demonstrations of model-free approach to control on a variety of problems, only time will tell if it is indeed a safe approach in every operating condition. As the NASA X-15 program taught us that performance does not imply stability. The following two questions (and their derivatives) remain unanswered at this time:

(*i*) Is there a provably-correct stable machine learning control (that is different from adaptive control)?

(*ii*) In the case of model-free approach, what are the definitions of state or stability? More, fundamentally is there a need for such paradigms?

4 Conclusion

In this note, we considered the question of stability in a model-free approach to control. Specifically, we first present an overview of traditional control concepts with specific focus on the issue of stability and its related concepts such as state and state-space models. This is followed by a very high level introduction to machine learning techniques for control and discussed the difficulty as well as need for the concept of stability in such a model-free approach.

References

1. Alpaydin, E.: Introduction to Machine Learning. MIT Press, Cambridge (2014)
2. Antsaklis, P.J., Michel, A.N.: Linear Systems, vol. 1. Birkhäuser, Boston (2006)
3. Aström, K.J., Murray, R.M.: Feedback Systems: An Introduction for Scientists and Engineers. Princeton University Press, Princeton (2010)
4. Åström, K.J., Wittenmark, B.: Adaptive Control. Courier Corporation, Mineola (2013)
5. Bertsekas, D.P., Tsitsiklis, J.N.: Neuro-Dynamic Programming, 1st edn. Athena Scientific, Belmont (1996)
6. Bhatia, N.P., Szegö, G.P.: Dynamical Systems: Stability Theory and Applications, vol. 35. Springer, Heidelberg (2006). https://doi.org/10.1007/BFb0080630
7. Chellaboina, V., Bhat, S.P., Haddad, W.M.: An invariance principle for nonlinear hybrid and impulsive dynamical systems. Nonlinear Anal. Theory Methods Appl. **53**(3), 527–550 (2003)
8. Dullerud, G.E., Paganini, F.: A Course in Robust Control Theory: A Convex Approach, vol. 36. Springer, New York (2013). https://doi.org/10.1007/978-1-4757-3290-0
9. Duriez, T., Brunton, S.L., Noack, B.R.: Machine Learning Control-Taming Nonlinear Dynamics and Turbulence. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-319-40624-4
10. Dydek, Z.T., Annaswamy, A.M., Lavretsky, E.: Adaptive control and the NASA X-15-3 flight revisited. IEEE Control Syst. **30**(3), 32–48 (2010)
11. Fogel, D.B.: System Identification Through Simulated Evolution: A Machine Learning Approach to Modeling. Ginn Press, Needham Heights (1991)
12. Ge, S.S., Hang, C.C., Lee, T.H., Zhang, T.: Stable Adaptive Neural Network Control, vol. 13. Springer, New York (2013). https://doi.org/10.1007/978-1-4757-6577-9
13. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
14. Haddad, W.M., Chellaboina, V.: Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach. Princeton University Press, Princeton (2008)
15. Haddad, W.M., Chellaboina, V., Nersesov, S.G.: Impulsive and Hybrid Dynamical Systems. Princeton Series in Applied Mathematics (2006)
16. Hovakimyan, N., Cao, C.: $\mathcal{L}1$ Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation. SIAM, Philadelphia (2010)
17. Ioannou, P.A., Sun, J.: Robust Adaptive Control, vol. 1. PTR Prentice-Hall, Upper Saddle River (1996)
18. Jenkins, D.R.: Hypersonics before the shuttle: A concise history of the X-15 research airplane (2000)
19. Lewis, F.L., Liu, D.: Reinforcement Learning and Approximate Dynamic Programming for Feedback Control, vol. 17. Wiley, New York (2013)
20. Ljung, L.: System identification. In: Procházka, A., Uhlíř, J., Rayner, P.W.J., Kingsbury, N.G. (eds.) Signal Analysis and Prediction. Applied and Numerical Harmonic Analysis, pp. 163–173. Springer, Heidelberg (1998). https://doi.org/10.1007/978-1-4612-1768-8_11
21. Michalski, R.S., Carbonell, J.G., Mitchell, T.M.: Machine Learning: An Artificial Intelligence Approach. Springer, Heidelberg (2013)
22. Narendra, K.S., Annaswamy, A.M.: Stable Adaptive Systems. Courier Corporation, Mineola (2012)

23. Ng, A.Y., Coates, A., Diel, M., Ganapathi, V., Schulte, J., Tse, B., Berger, E., Liang, E.: Autonomous inverted helicopter flight via reinforcement learning. In: Ang, M.H., Khatib, O. (eds.) Experimental Robotics IX. Springer Tracts in Advanced Robotics, vol. 21, pp. 363–372. Springer, Heidelberg (2006). https://doi.org/10.1007/11552246_35
24. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, vol. 1. MIT press, Cambridge (1998)
25. Sutton, R.S., Barto, A.G., Williams, R.J.: Reinforcement learning is direct adaptive optimal control. IEEE Control Syst. **12**(2), 19–22 (1992)
26. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco (2016)
27. Zhou, K., Doyle, J.C.: Essentials of Robust Control, vol. 104. Prentice hall, Upper Saddle River (1998)

Computational Core for Plant Metabolomics: A Case for Interdisciplinary Research

Vikram Pudi[1](✉), Pratibha Rani[1], Abhijit Mitra[1], and Indira Ghosh[2]

[1] IIIT Hyderabad, Hyderabad, India
vikram@iiit.ac.in
[2] JNU, New Delhi, India

Abstract. Computational Core for Plant Metabolomics (CCPM) is a web-based collaborative platform for researchers in the field of metabolomics to store, analyze and share their data. Metabolomics is a newly emerging field of 'omics' research that is concerned with characterizing large numbers of metabolites using chromatography, mass spectrometry and NMR. There is abundant volume and variety in the data, with velocity being unpredictable. An interdisciplinary engagement such as this faces significant non-technical challenges solvable using a balanced approach to software management in a university setting to create an environment promoting collaborative contributions. In this paper we report on our experiences, challenges and methods in delivering a usable solution. CCPM provides a secure data repository with advanced tools for analysis including preprocessing, pretreatment, data filtration, statistical analysis, and pathway analysis functions; and also visualization, integration and sharing of data. As all users are not equally IT-savvy, it is essential that the user interface is robust, friendly and interactive where the user can submit and control various tasks running simultaneously without stopping/interfering with other tasks. In each stage of its pipeline architecture, users are also allowed to upload external data that has been partially processed till the previous stage in other platforms. Use of open source softwares for development makes the maintenance and development of our modules easier than the others which depend on proprietary softwares.

1 Introduction

The metabolome [5] is the complete complement of all small molecule (< 1500 Da) [18] metabolites found in a specific cell, organ or organism. While the discipline of metabolomics has continued to develop as the comprehensive study of metabolism under genetic and environmental perturbations, with the advent of advanced analytical techniques, its usage and scope of application has undergone a paradigm change in recent times. Together with the genome, the transcriptome and the proteome, the metabolome is considered to be the fourth 'omic' building block of systems biology. Constituting the proverbial 'last mile' in the 'genotype to phenotype' paradigm, the metabolome encompasses perhaps the largest number of 'clues' for understanding the molecular basis of phenotypes.

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 223–234, 2017.
https://doi.org/10.1007/978-3-319-72413-3_15

The study of metabolomics enables researchers to gain precise insights into the mechanisms of nutrition and disease. When applied to plants, it helps develop new crop varieties that are resistant to pests and diseases and have better nutritional profiles and shelf life. It can help characterize the precise differences between multiple variants of a crop in terms of the metabolic pathways (biochemical reactions) affected. Hence, it can potentially throw light on the exact differences between a healthy crop and a diseased one, a resistant crop and a vulnerable one, or a genetically modified crop and a regular one, so as to examine the unintentional "side-effects" of a mutation. Clearly, the study of metabolomics has potential for revolutionizing the study of biological *systems*.

To cater to the need of a collaborative platform for the storage, analysis and sharing of metabolomics data, we have developed a web-based platform – Computational Core for Plant Metabolomics (CCPM). In this paper we report on our experiences, challenges and methods in delivering a usable solution for this inter-disciplinary engagement.

Today, data analytics techniques are mature – fast, accurate and scalable with several commercial and free implementations available. Scaling data analytics *across domains* remains the major challenge for the data analytics community. It is hence valuable and instructive for the community to identify the challenges and methods needed to deliver usable solutions for inter-disciplinary engagements. These include several non-technical challenges that we identify in this paper, in addition to technical challenges specific to the domain.

Specific technical challenges in our case include catering to: (i) a variety of metabolites, instruments & data formats, (ii) a variety of routine tasks with role based access, (iii) collaboration between several labs, and (iv) security and privacy concerns of users' data. To encounter these challenges, we have developed a unique end-to-end metabolomics analysis pipeline in CCPM. Open source softwares including some open bioinformatics specific packages have been used to build the platform. CCPM provides a secure repository called Laboratory Information Management System (LIMS) and analysis pipeline for metabolomics data and associated information. Following a modular architecture, separate modules have been designed for preprocessing, pretreatment and statistical analysis integrated with visualization tools.

Other available metabolomics data analysis platforms include XCMS [17], MetDat [8], MzMine [16], MetAlign [13], MetaboAnalyst [19], MetATT [20], Galaxy WorkFlow4metabolomics [11], etc. CCPM provides the following exclusive features: (1) Option of bulk data upload (2) Parallel execution of different tasks (3) Data format conversion and data compatibility with other existing formats (4) Option for analyzing externally preprocessed data obtained from other platforms (5) Data security features (6) Option of role assignment for various users in a project (7) Feature to handle long running processes on web.

2 Non-technical Challenges

As mentioned in the Introduction, scaling data analytics *across domains* remains the major challenge for the data analytics community. The bottle-neck in this endeavour lies in developing a deep and clear understanding of the *needs* of the domain, along with knowledge of the *possibilities* of IT-based solutions. This boils down to the largely non-technical challenges of effective *communication* between IT-savvy developers and IT-illiterate domain experts. This is neither easy nor common. Nor is it easy or common to have both skills in the same individual.

There is a resistance in the academic community to embrace the *no-man's land* of inter-disciplinary research [7]. Most admit that it is hard to delve deep enough into the separate disciplines in order to produce meaningful results. Domain experts look for usable solutions while computer scientists (CS) look for applications of their independent research. When domain experts are asked to describe their requirements, we often find two kinds of requests:

1. Tasks that are laborious for the expert, but easy to automate. These tasks are not of interest to academic CS researchers, as they are perceived as solvable with *just development* without research.
2. Tasks that are easy for the expert, but difficult to automate. These include AI-complete tasks requiring human cognitive abilities. If they are repetitive, current technology can attempt solutions, but often they are not repetitive and so the cost of developing automated solutions for numerous cases far exceeds the cost of manual effort.

Effectively, the tasks either appear trivial or impossible and hence CS researchers are generally disinclined towards interdisciplinary research. Unfortunately, this is an over-sight. Routine developers do not have the skills of *abstraction* necessary to deduce common structures of thought and capture the intentions and *implicit* needs of users in a complex academic domain.

Abstraction is a skill practiced by researchers. The same skill necessary for discovering beautiful theories is what is necessary for building simple and elegant software. In fact, those skills are needed in full measure – a good developer should not only be able to abstract out details when required, but also plunge into details when required. Getting stuck in details can change high-level plans and designs to finally result in more elegant, practical and maintainable systems. Just like a good theory, a simple and elegant software seems obvious and natural, *after* it has been created, not before.

Hence, interdisciplinary study needs domain experts, developers and researchers in a tight collaboration. This combination of skills may best be found in an academic, university setting.

3 Non-technical Challenges in a University Setting

Even though a university setting is the most amenable for interdisciplinary study, there are several common and identifiable challenges to be circumvented in such a setting so that such a study becomes possible and effective. These challenges mainly arise due to the lack of professional software development practices in a university setting, and are described in this section. This is not easily solvable because the incorporation of rigid software development guidelines in a university setting would hinder the flow of creative thought necessary for fluid abstraction, which is required for interdisciplinary research. A balanced approach is needed and in this section we also describe the methods that worked for us, which we feel did not burden the team members with excess and rigid project management just for the sake of feeling in control.

3.1 Continuous Re-development

The manpower available in good universities for software development consists mostly of students, who even when highly capable, work in the project along with other academic pressures, for a duration of about 1–3 years. This means that new students need to be continually trained in the conventions and standards of the particular software development project, further reducing the productive student life-span in the project. Also, budding developers are primarily concerned with getting desired features to work, and not particularly concerned about the readability and maintainability of code.

In this scenario, we often find that the system or module developed by a team of students cannot be easily built upon by subsequent teams because it is unreadable and unmaintainable. It is far easier for new teams to re-build the existing system or module from scratch. This leads to perpetual cycles of re-development whenever new requirements arise for a software project.

In our case, we averted this problem by ensuring that the development team contained students across batches, so that when some graduate others are still available to continue development and transfer knowledge to new recruits.

3.2 No Integration

It is common to break up a large project into independent modules to be developed separately by different teams with the hope of integrating them together later. In reality, the different modules/features would turn out to be hard to integrate. The teams working on them would develop different conventions and work almost till the end of the academic semester/year leaving little time and incentive to integrate their modules. Having created separate academic projects for students, there is no well-defined structure to ensure that the separate teams meet until the grading season. This results in large scale duplication of work, and errors or omissions.

This results in each team blaming the other for not having provided the requirements clear enough, or not understanding the requirements clearly.

In our case, we averted this problem by religiously conducting weekly project meetings including all members across the development and domain teams. This ensured that all members are always on the same page. This process is not without drawbacks, for it is common for some teams to discuss matters that are irrelevant to others, and hence seems a waste of time to them.

But nevertheless this cost of time is well spent. Even when one team does not understand everything discussed by the other, it provides opportunity to become familiar with the terms and ideas that the other teams are discussing. This *latent learning* is reinforced over several sessions. Eventually, this matures to the extent that the discussions and terms are meaningful enough that the development team can understand the *intentions* and *implicit needs* of the users, and the domain teams can understand the *possibilities* and *limitations* of what can be developed.

3.3 Closed System

Inspite of well-meant intentions of transparency, there is usually no process set in place for project investigators or users to initiate new feature development, fix problems with existing ones, or file complaints. There is usually no publicly available and monitored mailing list or call centre for such purposes. For many issues, people lose confidence that they will be fixed even if they complain.

In our case, we averted this problem by consciously setting up processes of collecting bug reports and feature requests from users, creating task lists and following up on them. The members of all teams are permitted to add to a common moderated task list. Regular quarterly workshops were also conducted to collect feedback from prospective end-users, and the feedback items were carefully interpreted and added to the common task list. Workshop participants were permitted to participate either physically or using online forums.

3.4 No Clear Standards

Usually, there are no best practices or detailed common conventions set in place for students to follow while developing and writing code. It is not clear what platforms to use or prefer, what processes to follow, and what documentation to produce. This results in non-uniform quality. Unfortunately, the quality of the system as a whole is determined by the strength of its weakest link, and so the resulting systems are usually non-maintainable, sub-optimal and contain ad-hoc code.

In our case, we averted this problem largely by using python, which is known to be a highly readable language. In-line documentation was therefore mostly unnecessary, and was added only when code needed to be changed to accommodate a new feature or solve a bug. Although inline documentation was mostly unnecessary, sufficient documentation was created to describe requirements by insisting that the domain teams communicate their finalized requirements of each module in writing without which the development team will not begin development of that module, even if the working had been explained and understood orally.

This had several positive impacts. The domain team would be able to crystallize their requirements by this process. The development team would be assured that there wouldn't be too many changes to the finalized requirements. The requirements documents created in this way serve to document the overall vision and direction of the project that can be reviewed and revisited whenever there is a conflict. The rigidity of insisting on requirements documents was only at the overall module level. Day-to-day changes would be discussed orally during weekly meetings, and added to the common task list without unnecessarily detailed descriptions.

The collection of task lists and requirements documents forms the bulk of documentation for the project. Additionally, a few documents were created to describe some coding standards (such as naming conventions), how to use GIT based versioning, to describe the different machines and platforms used, and a standard operating procedure (SOP) to document the processes to be followed by end-users and to describe what they can expect from the software system.

3.5 Not Thoroughly Tested

Testing is usually left to developers (students) only, whose primary concern many times is to only produce a working prototype that can be graded satisfactorily. Hence, there is no added incentive for all features to be tested adequately, in a time consuming, laborious manner. So, bugs typically remain and are not promptly fixed, and become hard to fix in the next academic cycle. This leads again to the first challenge above of continuous re-development.

In our case, we averted this problem by ensuring that domain team members test every feature and page on a development platform, before the code is pushed onto the main platform which is used by end-users. The fact that a non-developer tests regularly ensures that the final system is usable without separate or extensive alpha and beta testing. The testing is done by the domain team members by actually using the system in realistic scenarios on real datasets. While this doesn't ensure that every use case is tested, it is sufficient for usability. Any lurking bugs will be dealt with as and when they are reported. Only critical parts of the code dealing with security and stability of data go through an expensive process of thorough testing of all use cases.

3.6 Strengths of Our Approach

By having only a few developers at a time, who are all familiar with the full stack of development (including analysis, design, UI, coding and deployment) and following the processes described above, we were able to ensure that the developers become familiar and empathetic with the intentions and implicit needs of users. This was the major strength that led to correct priorities between all aspects, including usability, development time, development cost, efficiency and resources. The domain teams were instructed not to describe and micro-manage how they want the system to work, but to simply convey the intention of what they want. It was suggested that this would give freedom to the developers to build something beyond their imagination, and it often worked that way. The result was a complete usable system, and we believe these simple processes are reproducable in other settings requiring interdisciplinary application of data analytics in other domains.

4 Overview of CCPM

Computational Core for Plant Metabolomics (CCPM) is the result of the application of the principles and processes described in the previous section to develop a platform for the storage, analysis and sharing of metabolomic data. As mentioned in the Introduction, specific technical challenges in our case include catering to: (i) a variety of metabolites, instruments & data formats, (ii) a variety of routine tasks with role based access, (iii) collaboration between several labs, and (iv) security and privacy concerns of users' data.

Additionally, among the end-users which are labs that produce metabolomics data, the majority are small and medium sized, and do not have the technical resources and manpower to host and administer a data centre of their own. This is the typical scenario for most interdisciplinary projects as they deal with domain experts who are not skilled in IT aspects. Hence, the choice of platform to implement such projects is a web-based one with a common data centre maintained centrally. The end-users will have the convenience of using the platform, using a familiar browser interface, without the hassle of administering it.

To implement our CCPM platform, we have used web2py [15], an open-source full-stack python-based web application development framework. Web2py is a model-view-controller (MVC) framework that allows clear separation of concerns of the model, logic, and UI, resulting in easy to read, maintainable code. It comes with implementations of strong security features, permitting us to focus on defining role-based access alone to complete the security requirements.

Following a modular architecture, separate modules have been designed for data upload, preprocessing, pretreatment and analysis integrated with visualization tools. For the entire development we have used open source softwares and open platforms including HTML5, SQLite, MySQL and javascript along with some R-language based bioinformatics packages like NetCDF [6], gplots [4] and KEGGREST [12]. We also use bootstrap for building a modern and intuitive GUI, packaging the complexity of an entire task scheduling workflow in a deceptively simple interface.

Use of open source and open standards goes hand in hand with the philosophy of collaborative interdisciplinary research and development, as: (1) It is free to use, modify and distribute. (2) Its working and security is more trusted as the code is accessible to everyone. It is continually evolving and anyone can fix bugs as they are found, without users having to wait for the next release. (3) It uses open standards accessible to everyone; thus, it does not have the problem of incompatible formats that exist in proprietary software. (4) Lastly, the platforms using open-source software do not have to think about complex licensing models and do not need anti-piracy measures like product activation or serial number. We chose to use GitLab [3] to facilitate the development process. It provides easy to use options of version control and software maintenance for developers.

While developing such a platform the framework should be flexible and should allow for easy and straightforward development of new data processing modules. We address this by keeping a strict separation between the application core and individual modules for data processing and visualization. To cope with the heavy user load, the CCPM v4.4 server is currently hosted and accessible on two dedicated machines (one at IIIT-Hyderabad, India [2] and one at JNU, India [1]) that are maintained and updated regularly.

5 Features and Functionalities

In this section, we describe the various features, functionalities and describe our approach in the design and implementation of CCPM's various modules.

5.1 Data Formatting and Standardization

The management, storage and standardization of metabolomics data is absolutely critical to making metabolomics properly integrated into the other 'omics' sciences [9]. Another critical aspect is to make instrumental data uniformly readable and readily exchangeable. This is particularly challenging since metabolomics data can be collected by a far larger variety of instruments coming from a wider range of manufacturers than for data for other 'omics'. We have adopted NetCDF (Network Common Data Form) [6] and ANDI (ANalytical Data Interchange protocol) [14] to solve the format mismatch and data conversion issues. This provides a solution for handling multi-lab or multi-investigator projects and in bringing some semblance of uniformity to input data. We use MSI [10] standard of data architecture to store and analyze the data.

5.2 Data Upload Module with Bulk Data Upload Option

Metabolomics data is obtained as the output of the processing of chromatography, mass spectrometry or NMR instruments on specific biological samples of interest. The metadata stores the complete record of instrumental parameters and storage information of biological samples to tag the instrument's output for the sample.

The data upload module provides facilities for both bulk data and single file upload along with metadata entries of individual sample files. The bulk upload module uses a queuing method to provide the facility for uploading multiple files together. Bulk upload resumes from the point of interruption if any interruption happens in-between. Bulk upload is essential when dealing with large numbers of samples, and especially when transferring data from another platform into CCPM.

CCPM initially provides 5 GB space to each user. This space can be extended on user request. Currently netCDF, mzXML, mzML, mzData file formats are accepted. Phenotypic data from the barcode reader in a excel file can also be uploaded. Through uploading this file, the values will automatically populate into the phenotype forms of the samples. The user can also download the "Phenotype template".

5.3 Pre-processing Module

This module is responsible for feature detection, comparative analysis and visualization of preprocessed results. It provides the facility to download the preprocessed result files individually or in a consolidated form as a compressed ".rar" file. The CCPM pipeline architecture also provides an option for analyzing externally preprocessed data obtained from other platforms or instruments which the other existing platforms do not provide. This is an essential feature as several instruments output data that has already been preprocessed.

5.4 Data Analysis and Visualization Module

This module provides functions for performing various kinds of data analysis and visualizing the results. The purpose of this analysis is to determine the metabolites that are present in a sample, or which are significantly different in the sample compared to other samples. The kinds of analysis available in the module include univariate analysis, fold change across 2 groups of samples, correlation analysis, t-tests, volcano plots, clustering analysis, heatmaps, self-organizing maps, PCA, SVM and PLS-discriminant analysis. At each step of the pipeline execution, the results are saved so that the user can re-run the tasks without needing to start from scratch. User can also download the intermediate results.

5.5 KEGG Connectivity Module

In this module, we provide the facility to connect to the KEGG [12] database through its API using KEGGREST [12]. The KEGG database contains data of metabolomic pathways, which is the set of chemical reactions and their pathways across several biological species. Connecting to this database allows users to determine the metabolomic pathways that are affected by chosen metabolites and to determine those pathways that are up-regulated or down-regulated in a sample variety.

5.6 User Roles

CCPM provides different levels of accessibility to the projects for different users. This provides a LIMS (laboratory information management system) like hierarchy and added security to the project and its data. The type of modifications that can be done by a user depends on the rights provided to him in a project by the PI of that project. A user of the CCPM platform can be allotted any of the roles given below:

1. **Principal Investigator (PI):** They have permission to add new projects and publish project data. They can also grant roles to the members who want to join a project.

2. **Experimentalist:** To be an experimentalist, user should join the project first with permission from the project's PI. Once their role is accepted by the PI/Coordinator, they can create sample groups, add sample data into those groups and perform analyses on that data.
3. **Co-investigator (Co-PI):** The Co-investigator should first join the project with permission from the project's PI. Co-investigators can perform analyses in the projects they have enrolled in and can view the results. However, they cannot upload/modify any data in the project.

Additionally, we have a role for regular users, who can signup, login, explore and analyze available published data, but cannot add or modify any data.

5.7 Security Features

Web2py comes with implementations of strong security features, permitting us to focus on defining role-based access alone to complete the security requirements. However, web2py has a web-based administrative interface to access data. So anyone with administrative permissions on the system hosting web2py can access data. To prevent this, we have implemented security measures to make key tables inaccessible via the administrator panel, while allowing others to be accessed for convenience. Accordingly, relevant web2py code has been modified so that the data of these excluded tables is protected from view/insert/update/delete operations.

Effective sanitization has been implemented so that even if the administrator attempts to manipulate URL in order to access contents of restricted tables, the access is denied. Thus even the admin of the system can't use any PI's data and projects. By effectively implementing role-based access one PI can't see or access other PI's data and projects. Even Co-PI can't see or access his PI's data or project without proper authorization by his PI. Co-PI and Experimentalists under a PI can access only those projects and data for which their PI has authorized them. These users have an expiry date of one year which needs renewal by the concerned PI.

Additionally, a PI can request to withdraw his published project. In that case, the relevant project is marked with a "withdrawn" tag in the database and is not physically visible to anybody but continues to reside on the hard disk.

5.8 Long Running Processes

The web doesn't directly allow running of long running processes due to potential non trustworthiness of connections available between the user and the server. Hence, any webpage that consumes too much time results in a *timeout* error. However, preprocessing and data analysis tasks are typically long running processes and consume significant computational resources.

To overcome this problem, in our design we have architected an entire task scheduling and management workflow into CCPM's design, built upon web2py's implementation of a task scheduler to run background processes. Using this, users submit preprocessing or analysis tasks on the CCPM portal and manage a list of their running and completed tasks. By this mechanism, the web-page returns immediately after scheduling the long-running task, instead of waiting for it to finish. Once the task is completed, the result is displayed in the user's task list, which he can return to at any time. The task submitted can be found under a "Running Tasks" tab of the user. The status of the task is displayed (queued, assigned, running or failed) in the task list. A "Refresh" button is shown next to the task name to refresh its status. An option to delete tasks is also provided.

None of the other currently existing online metabolomics platforms provide this facility of task management and for long running processes, which is essential in practice for preprocessing and analysis tasks.

5.9 Intuitive User Interface (GUI)

We have designed the platform layout to show available options in a clutter-free manner, and to ease the accessibility of information & services. For example, all the available projects are grouped at one place. All the tasks pertaining to a user are grouped together in a separate tab. All the required tools and available functionalities are in the "tools" section but in separate tabs. The portal provides a user friendly interactive interface where the user can control various tasks running simultaneously without stopping /interfering with other running tasks. We also provide the option of selecting and re-running old tasks with changed parameters, and store the new tasks separately. All of this complexity is hidden behind a deceptively simple and intuitive interface.

6 Conclusions

After studying various existing platforms available for analyzing metabolomics data we developed CCPM – an end-to-end platform to provide the metabolomics research community with a collaborative platform to store, analyze and share their data. CCPM provides several novel facilities including: (1) bulk data upload (2) task scheduling and management of long running processes (3) data format conversion and data compatibility with other existing formats, and (4) analyzing externally preprocessed data obtained from other platforms. Additionally, the platform provides a collaborative environment with role-based access and data security. Being an interdisciplinary R&D effort, significant non-technical challenges were faced and dealt with using a balanced approach to software management in a university setting to create an environment promoting collaborative contributions.

Acknowledgements. This work was supported by a research grant from the Department of Biotechnology, Govt. of India.

References

1. Computational core for plant metabolomics: Jnu platform. http://metabolomics.jnu.ac.in/lims. Accessed 10 Oct 2017
2. Computational core for plant metabolomics: Main platform. http://metabolomics.iiit.ac.in/user/login. Accessed 10 Oct 2017
3. GitLab. https://about.gitlab.com/. Accessed 10 Oct 2017
4. gplots: Various R programming tools for plotting data. https://cran.r-project.org/web/packages/gplots/index.html. Accessed 10 Oct 2017
5. Metabolome. https://en.wikipedia.org/wiki/Metabolome. Accessed 10 Oct 2017
6. RNetCDF: Interface to NetCDF datasets. https://cran.r-project.org/web/packages/RNetCDF/index.html. Accessed 10 Oct 2017
7. Mind meld. Nature Editorial, 525(7569), September 2015
8. Biswas, A., Mynampati, K.C., Umashankar, S., et al.: MetDAT: a modular and workflow-based free online pipeline for mass spectrometry data processing, analysis and interpretation. Bioinformatics **26**(20), 2639–2640 (2010)
9. Fiehn, O., Kristal, B., Ommen, V., et al.: Establishing reporting standards for metabolomic and metabonomic studies: A call for participation. OMICS A J. Integr. Biol. **10**(2), 158–163 (2006)
10. Fiehn, O., Robertson, D., Griffin, J., et al.: The metabolomics standards initiative (MSI). Metabolomics **3**(3), 175–178 (2007)
11. Giacomoni, F., Le Corguille, G., Monsoor, M., et al.: Workflow4Metabolomics: a collaborative research infrastructure for computational metabolomics. Bioinformatics **31**(9), 1493–1495 (2015)
12. Kanehisa, M., Goto, S.: KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res. **28**(1), 27–30 (2000)
13. Lommen, A.: MetAlign: interface-driven, versatile metabolomics tool for hyphenated full-scan mass spectrometry data preprocessing. Anal. Chem. **81**(8), 3079–3086 (2009)
14. Matthews, L., Miller, T.: ASTM protocols for analytical data interchange. JALA J. Assoc. Lab. Autom. **5**(5), 60–61 (2000)
15. Di Pierro, M.: web2py for scientific applications. Comput. Sci. Eng. **13**, 64–69 (2011)
16. Pluskal, T., Castillo, S., Villar-Briones, A., Oresic, M.: MZmine 2: modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data. Bioinformatics **11**(395) (2010)
17. Smith, C.A., Want, E.J., OMaille, G., Abagyan, R., Siuzdak, G.: XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. Anal. Chem. **78**(3), 779–787 (2006)
18. Wishart, D.S.: Current progress in computational metabolomics. Brief Bioinform. **8**(5), 279–293 (2007)
19. Xia, J., Psychogios, N., Young, N., Wishart, D.S.: MetaboAnalyst: a web server for metabolomic data analysis and interpretation. Nucleic Acids Res. **37**(Web Server issue), W652–W660 (2009)
20. Xia, J., Sinelnikov, I., Wishart, D.S.: MetATT: A web-based metabolomics tool for analyzing time-series and two-factor datasets. Bioinformatics **27**, 2455–2456 (2011)

Semantic Interoperability in Electronic Health Record Databases: Standards, Architecture and e-Health Systems

Subhash Bhalla[1], Shelly Sachdeva[2](✉), and Shivani Batra[2]

[1] Division of Information Systems, University of Aizu,
Aizu-wakamatsu, Fukushima 965-8580, Japan
bhalla@u-aizu.ac.jp

[2] Department of Computer Science and IT,
Jaypee Institute of Information Technology University, Noida 201301, India
sachdevashelly1@gmail.com, ms.shivani.batra@gmail.com

Abstract. Information systems have been deployed in different clinics and hospitals to preserve patient data. In order to promote the exchange of data among systems (and organizations), standards are being adopted for data exchange. Further, the clinics and hospitals aim to manage a patient's life-time history of records. A piece of the individual patient's medical record can be captured, stored, queried, and shared over a network through enrichment in information technology. Thus, electronic health records (EHRs) are being standardized for incorporating semantic interoperability. In addition, a generic storage structure is required to capture distinguished data requirements of various organizations. The generic structure must be capable of dealing with sparseness and frequent evolution behavior of EHRs. A subsequent step requires that healthcare professionals and patients get to use the EHRs, with the help of technological developments, such as workflow toolkits and new (easy) query languages. The goal is to present an overview of different approaches in understanding some current and challenging concepts in e-health informatics. Successful handling of these challenges will lead to improved quality in healthcare by reducing medical errors, decreasing costs, and enhancing patient care. The report is focused on the following objectives: (1) understanding the role of EHRs Databases; (2) understanding the need for standardization to enhance quality; (3) establishing interoperability in maintaining EHRs; (4) explicating a framework for standardization and interoperability (the openEHR architecture); (5) exploring various data models for managing EHRs; and (6) understanding the difficulties in querying data in EHR and e-health systems.

1 Overview and Motivation

Healthcare is an information-intensive activity. It produces large quantities of data from laboratories, wards, operating theatres, primary care organizations, and from wearable and wireless devices [16]. Recently, Electronic Healthcare Records (EHRs) are gaining popularity due to its application in healthcare domain [1, 15]. EHRs are becoming a method by which physicians are able to electronically capture high-quality data at a fast

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 235–242, 2017.
https://doi.org/10.1007/978-3-319-72413-3_16

speed and at low cost. The EHRs have specialized structure, sparseness and volatility that requires extensive research [2]. Many researches discuss database research from various perspectives [13]. It can be applied to deal with standardized EHRs to provide efficient care and reduce costs.

1.1 Role of EHRs

An Integrated Care EHR [17] is defined as: “a repository of information regarding the health of a subject of care in computer processable form, stored and transmitted securely, and accessible by multiple authorized users. It has a commonly agreed logical information model which is independent of EHR systems. Its primary purpose is the support of continuing, efficient and quality integrated healthcare and it contains information which is retrospective, concurrent and prospective”. EHRs contain the longitudinal health history of each patient. It is required to improve quality of care. EHRs are widely exploited in telemedicine, emergency situations, homecare, epidemiological situations, and in creating an e-health environment. They help to prevent medication errors, reduce duplication, and save time. It will facilitate better coordination of long-term patient data. Hence, EHRs must be designed to capture relevant clinical data using standardized data definitions and standardized quality measures. These will help in improving preventive care and in increasing physician efficiency. Some of the main concerns in maintaining EHRs are standardization, interoperability, privacy and security.

1.2 Interoperability in Maintaining EHRs

Information sharing across medical institutions is restricted to information exchange between specific partners. There is a need to achieve timely and consistent access to EHRs. Currently, there is no single universally accepted clinical data model that will be adhered by all [2]. The meaning of information must be preserved across various applications, systems, and enterprises. It is essential to support interoperability between software from different vendors. Another major problem is the huge amount of different (proprietary or standardized) interfaces that are in use. The current study discusses the model (two-level model) and specifications (based on EHR standard) for achieving interoperability.

2 Scope and Structure

The current report aims to cover the background information related with the Standardized EHRs. The modeling approaches and the research activity will be presented. Various open source alternatives for creating information systems will be presented. The aim is to focus on following topics:

1. Standards for messages, concepts and storage systems,
2. Data modeling for EHRs,
3. Handling archival data,

4. Querying EHRs data,
5. On-going research, standards and prototypes,
6. Open research problems, and
7. Security and privacy concerns.

3 Data Modeling for Electronic Health Records

3.1 Two-Level Modeling Approach

In order to achieve interoperability, a two-level modeling approach for separation of information and knowledge is specified in open electronic health record architecture (EHRA) [14, 20, 21]. The two-level approach consists of a reference model (RM) and a conceptual model (the domain-level definitions in the form of archetypes and templates). The concept behind it is the introduction of a level of abstraction between the program logic and the database schema. This mechanism provides data independence, similar to the case of conventional database management systems (DBMSs) [2]. EHR systems based on this approach have the capability of incorporating new standardized data elements in a timely manner. A domain expert designs archetypes, and the user creates the information item which is mapped to an archetype [18]. The dual model EHRA specifications initially proposed by openEHR [27] have already been adopted by Microsoft [Microsoft Health] [26]; and HL7 [22].

A conceptual definition of data as archetype can be developed in terms of constraints on structure, types, values, and behaviors of RM classes. Standardization can be achieved in following manner. Whenever there is a change in the clinical knowledge (or requirements), the software need not be changed. The archetypes need to be modified (or added) in conformity with RM. This leads to enhancement in terms of data quality and information quality.

Archetype Definition Language (ADL) syntax has been proposed by openEHR [27]. It is one possible serialization of an archetype. It is used to describe constraints on data which are instances of the reference model (information model). The Archetype Model structurally expresses the semantics of the ADL (see [2] for details). ISO has accepted ADL as a standard language for description of archetypes [21].

3.2 Standardized EHRs

Organizations adopt standards to achieve interoperability and promote information quality [25]. However, there are problems in reaching agreements on standards.

EHRs consists of different types of data (textual descriptions, numeric values, logical values, date and time expressions and hierarchical data structures) with new data requirements emerging with the passage of time. They have a complex structure that may include data from about 100 to 200 parameters, such as temperature, blood pressure, and body mass index. Individual parameters will have their own contents. Each contains an item such as “data” (e.g., captured for a blood pressure observation). It offers complete knowledge about a clinical context, (i.e., attributes of data); “state” (context for interpretation of data); and “protocol” (information regarding gathering of data).

The structure and content of the lifelong EHRs requires standardization efforts. In order to serve as an information interchange platform, EHRs aim to use archetypes [18, 21] to accommodate various forms of content. The contents may be structured, semi-structured or unstructured, or a mixture of all three. The main organizations/ standards working on semantic interoperable EHRs are openEHR, HL7 and CEN/ISO 13606.

3.3 Data Quality in EHRs

Data quality is important. It is considered within a (narrow) scope of data verification and validation. Data quality should also concern the important aspect of assuring that EHR data is appropriate and adequate for use. The critical data issues can be incompleteness (missing information), inconsistency (information mismatch between various sources or within the same EHR data sources), and inaccuracy (nonspecific, non-standards based, inexact, incorrect, or imprecise information). These types of inaccuracies in the attribute values of patient records make it tough to find specific patient care information for research and treatments. The data collected in various systems can have quality faults. For instance, it may be non-coherent or include contradictory information. The desired data may be completely missing. For example, the unit for weight may not be entered definitively as kilogram or pounds, or may be outside the permissible range. A patient's health information is shared in a multi-disciplinary (shared care) environment, therefore there is a need for a communication format and protocol for the purpose of standardization. Thus, the development and adoption of national and international standards for EHR interoperability is essential. Standardization will enhance the quality of EHR systems [2] and, in this regard, many research studies discuss different approaches to improve quality.

4 Handling Archival Data

With the prevalent acceptance of the EHRs by various health organizations across the globe, huge amount of health data is readily available. Consequently, there is an increasing need to utilize and manage this data to deliver quality healthcare. The OpenEHR artefacts (archetypes and templates) have a deeply nested structure and each concept has its own data nodes. The persistence layer for these EHRs needs to be capable to handle such a structure. The EHR data belonging to the OpenEHR standard's reference model (RM) can be serialized into several formats such as JSON, XML and others [14].

The OpenEHR forum [27], does not mention to use a specific persistence method for the archetypal EHRs. Wang et al. have designed and implemented an Archetype-driven Biomedical Data Platform on Relational Database (ABDP) which could achieve flexible data storage by Archetype Relational Mapping (ARM) [23]. On the basis of ARM, the ABDP implemented Archetype Query Language (AQL) [24] through a set of web services to provide flexible data access. Freire et al. have implemented an OpenEHR Data Platform on XML Database [12]. The need of storing the standardized EHRs data is not sufficed by XML-based persistence. The way the archetypes are designed and the nature of the data values that are stored in the database make the automatically generated indexes in the XML databases inefficient. Moreover, the tree structured archetypes are relatively deep and consists of repeated path segment identifiers. This requires the persistence layer to facilitate easy querying of these structures along with being capable to perform in-depth querying. The alternative may be to move from the traditional relational and XML databases to highly scalable, high-performance and schema-less databases known as NoSQL databases. Madaan et al. have implemented persistence level storage system for the archetypal EHRs using a NoSQL database (Mongo DB) [6]. This eliminates the object-relational mapping and maintains a node and path based persistence.

Considering consistency and availability as primary concerns, relational database management system (RDBMS) outperforms NoSQL database management system. Conventionally, RDBMS were designed to define, manipulate, query and control data stored in form of horizontal rows. The horizontal row format falls behind to capture constantly evolving and sparsely populated behavior of data. This demanded for a paradigm shift. Therefore, a variation to traditional row storage approach, namely EAV (Entity Attribute Value model) came to existence that proposed to store data in vertical columns [7]. EAV transcend to store high dimensional sparse data, but it lacks in search efficiency. To achieve search efficiency, surrogate models were proposed namely Dynamic Tables (DT), Optimized Entity Attribute Value (OEAV) and Optimized Column Oriented Model (OCOM), but none was observed to be superlative [8].

PolyEHR [3] is a framework which builds health application templates using archetypes and store EHR data in heterogeneous databases by means of polyglot persistence. It proposes to store the clinical EHR data that undergo frequent changes in flexible data models (e.g. key-value, document and graph), while other data that requires less alterations are stored in relational data schemas.

5 Querying EHRs Data

Considering healthcare domain, querying is important from two perspectives. The first is query over archival data of a patient, and the second is query over large population data for research studies.

5.1 Querying EHR Data Using Query Languages

With existing query languages (such as XQuery, SQL, OQL), users must know the persistent data structure of an EHR in order to write an appropriate query for querying EHR data. Thus, none of these can be directly used as a query language required by integrated care EHRs. Queries are expressed in a language that is a synthesis of SQL (SELECT/FROM/WHERE) and W3C XPaths, extracted from the archetypes. The language is called the archetype query language (AQL) [24].

5.2 High-Level Query Language Interfaces

Higher-level support is an active area of research. Many research efforts aim to improve user interaction facilities [4, 19]. This will improve the quality of care. Querying the system with the two-level model architecture is not the same as querying a relational database system or an XML database system. At the user level, querying data regarding "body mass index" (BMI) must be made very simple. The user only knows BMI as a parameter and will query that parameter only. There is a need for a query support that is neutral to system implementation, application environment, and programming language. Several high-level query languages such as QBE, QBO, XQBE, and XML-GL exist.

It is possible to convert specification (in ADL) and patient data into its equivalent form, presented through XML. One of the possible approaches for querying EHRs is to use ADL to generate a storable XML output of the corresponding XML database and then to use XQuery. Also, we can use XQBE on the top of the generated XML file [11].

AQBE addresses the challenges of querying EHRs data based on semantic interoperability, and provides an easy to use interface which can be used by skilled and semi-skilled users [5].

6 Security and Privacy Concerns

The requirements for security and privacy are critical and difficult to satisfy in case of EHRs data as compared to any other data. This is due to the conflicting needs of clinicians (who demand open and easy access to databases) and the patients (who prefer closed and private access to information stored in databases). Standards ensure the confidentiality, integrity, and availability. Basic security and privacy requirements of EHR systems are met using standard ISO/TS 14441 [10]. A model for EHRs database systems has been proposed which tried to include the security at each reference layer of the standardized EHRs [9]. A simulation of various techniques that can be applied to the each layer has been discussed.

7 Summary and Conclusions

Standardized EHRs are being adopted in large scale at US hospitals and in healthcare enterprises. Many research studies/projects are considering various ways of supporting e-Health systems. In addition to supporting the existing workflows in healthcare domain, the research must consider the query needs of healthcare workers and their patients. Similarly, medical science aims to conduct research on population data. In order to access the archives of life-time records of patients, the security and privacy concerns must be addressed by research studies.

References

1. Hsiao, C.J., Hing, E.: Use and Characteristics of Electronic Health Record Systems Among Office-Based Physician Practices, United States, 2001–2012 (2012)
2. Sachdeva, S., Bhalla, S.: Semantic interoperability in standardized electronic health record databases. J. Data Inf. Qual. (JDIQ) **3**(1), 1 (2012)
3. Araujo, A.M.C., Times, V.C., Silva, M.U.: Poly EHR: a framework for polyglot persistence of the electronic health record. In: International Conference on Internet Computing and Internet of Things, pp. 71–77 (2016)
4. Braga, D., Campi, A., Ceri, S.: XQBE (XQueryBy example): a visual interface to the standard XML query language. ACM Trans. Datab. Syst. **30**(2), 398–443 (2005)
5. Sachdeva, S., Yaginuma, D., Chu, W., Bhalla, S.: AQBE - QBE style queries for archetyped data. IEICE Trans. **95-D**(3), 861–871 (2012)
6. Madaan, A., Chu, W., Daigo, Y., Bhalla, S.: Quasi-relational query language interface for persistent standardized EHRs: using NoSQL databases. In: Madaan, A., Kikuchi, S., Bhalla, S. (eds.) DNIS 2013. LNCS, vol. 7813, pp. 182–196. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37134-9_15
7. Duftschmid, G., Wrba, T., Rinner, C.: Extraction of standardized archetyped data from electronic health record systems based on the entity-attribute-value model. Int. J. Med. Inform. **79**(8), 585–597 (2010)
8. Batra, S., Sachdeva, S.: Suitability of data models for electronic health records database. In: Srinivasa, S., Mehta, S. (eds.) BDA 2014. LNCS, vol. 8883, pp. 14–32. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13820-6_2
9. Mehndiratta, P., Sachdeva, S., Kulshrestha, S.: A model of privacy and security for electronic health records. In: Madaan, A., Kikuchi, S., Bhalla, S. (eds.) DNIS 2014. LNCS, vol. 8381, pp. 202–213. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05693-7_13
10. ISO/TS 14441:2013 Health Informatics – Security & Privacy Requirements of EHR Systems for Use in Conformity Assessment https://www.iso.org/standard/61347.html
11. Sachdeva, S., Bhalla, S.: Tutorial: implementing high-level query language interfaces for archetype-based electronic health records database. In: Proceedings of the International Conference on Management of Data (COMAD), pp. 235–238 (2009)
12. Freire, S.M., Sundvall, E., Karlsson, D., Lambrix, P.: Performance of XML databases for epidemiological queries in archetype-based EHRs. In: Scandinavian Conference on Health Informatics 2012, vol. 070, pp. 51–57, 2–3 October 2012
13. Abadi, D., et al.: The Beckman report on database research. Commun. ACM **59**(2), 92–99 (2016)
14. Beale, T., Heard, S.: The openEHR architecture: architecture overview. In: The openEHR release 1.0.2. openEHR Foundation (2008)
15. Dogac, A.: Interoperability in eHealth systems. Proc. VLDB **5**(12), 2026–2027 (2012)
16. Simonov, M., Sammartino, L., Ancona, M., Pini, S., Cazzola, W., Frascio, M.: Information, knowledge and interoperability for healthcare domain. In: Proceedings of the 1st International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS 2005). IEEE, Los Alamitos (2005)
17. ISO/TC 215 Technical report. Electronic health record definition, scope, and context. (2nd. draft, August) (2003)
18. Beale, T., Heard, S.: The openEHR archetypemodel: openEHR templates. In: openEHR release 1.0.2. (2009). Issue Date 20 April 2009

19. Jayapandian, M., Jagadish, H.V.: Automating the design and construction of query forms. IEEE Trans. Knowl. Data Eng. **21**(10), 1389–1402 (2009)
20. ISO 13606-1: Health Informatics: Electronic Health Record Communication. Part 1: Reference Model, 1st edn. (2008)
21. ISO 13606-2: Health Informatics: Electronic Health Record Communication. Part 2: Archetype Interchange Specification, 1st edn. (2008)
22. HL7. Health level 7. www.hl7.org. Accessed June 2017
23. Wang, L., Min, L., Wang, R., Lu, X., Duan, H.: Archetype relational mapping-a practical openEHR persistence solution. BMC Med. Inform. Decis. Mak. **15**(1), 88 (2015)
24. Archetype Query Language. https://openehr.atlassian.net/wiki/spaces/spec/pages/4915244/Archetype+Query+Language+Description. Accessed July 2017
25. Lewis, G.A., Morris, E., Simanta, S., Wrage, L.: Why standards are not enough to guarantee end-to-end interoperability. In: Proceedings of the IEEE 7th International Conference on Composition-Based Software Systems. IEEE, Los Alamitos (2008)
26. Microsoft Connected Health Framework. http://www.microsoft.com/industry/healthcare/technology/Health-Frameok.mspx. Accessed May 2017
27. OpenEHR Community. http://www.openehr.org/. Accessed May 2017

Big Data Analytics Advances in Health Intelligence, Public Health, and Evidence-Based Precision Medicine

Asoke K. Talukder(✉)

Precision Genomics Limited, Kowloon, Hong Kong
asoke@vibranthealthanalytics.com

Abstract. Large amount of patient data is available in hospitals. Moreover, a huge body of medical knowledge is available in digital form in the public domain, like NLM (National Library of Medicine), PubMed, NCBI (National Center for Biotechnology Information), MeSH (Medical Subject Heading), OMIM (Online Mendelian Inheritance in Man). There are also public biomedical databases like PDB (Protein Data Bank), GO (Gene Ontology), Chemical Entities of Biological Interest (ChEBI), KEGG (Kyoto Encyclopedia of Genes and Genomes), Drug databases (DrugBank), Recon (Reconstruction of Human Metabolism), dbSNP (DNA Mutation Database), COSMIC (Catalogue of Somatic Mutations in Cancer), etc. The list goes on and on and on. In this paper, we are addressing the challenge – how does our analytic solution combine these data and knowledge bodies through the technology of big-data combined with artificial intelligence, mathematical models, and translational medicine into "***Evidence Based Precision Medicine – the perfect decision outcome with perfect knowledge backing***." The benefits are immense for many stakeholders. Payer costs are reduced significantly – be it an insurance company or employer or an uninsured individual. The accuracy of medical decisions including the hospital productivity are increased significantly, with reduced medical errors, reduced disease burden, reduced fraud and wastage. Evidence based precision medicine will benefit patients, patients' families, doctors, hospitals, insurance companies, payers, Government regulators, healthcare professionals, public exchequers and finally, improve the overall general health of the population as a whole.

Keywords: Big data analytics in healthcare · Health intelligence
Clinical pathway · Public health · Evidence based precision medicine
Precision medicine · Triple aim · Better health better outcome at better price
Predictive healthcare · KPI · NGS · Genomics

1 Introduction

The human body is the most complex lab (laboratory) in the universe where all reactions take place at 37 °C and one atmospheric pressure. This lab is made up of many body parts or organs. These organs are made from different types of cells and tissues. If we take a deep dive further, we will observe complex synchronized activities

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 243–253, 2017.
https://doi.org/10.1007/978-3-319-72413-3_17

within a cell (intracellular) and outside of a cell (extracellular). Various biomolecules are constantly interacting with one another to keep us alive. Even there are complex signaling systems to start, stop or speed up a biochemical reaction. Sometimes, there are malfunctions of some components of the organs or the signaling system that manifest as disease and make us sick. This biological system is multi-scale, ranging from nano meters to a few thousands of kilometers as shown in Fig. 1. Pharmacology on the other hand includes mathematical models to quantify the outcome of a clinical trial. In pharmacology, the drug manufacturer is required to show that the drug is safe and will work on any individual. This is like certifying a computer program will work on any dataset irrespective of whether the program was tested with this data or not.

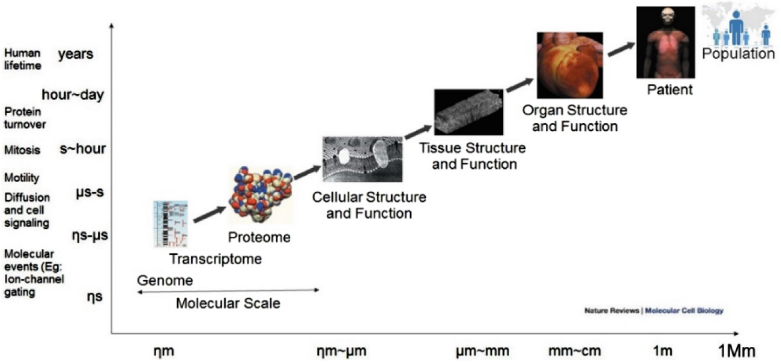

Human lifetime
years
hour~day
Protein turnover
Mitosis
s~hour
Motility
Diffusion and cell signaling
μs-s
ηs-μs
Molecular events (Eg: Ion-channel gating
ηs
Population
Patient
Organ Structure and Function
Tissue Structure and Function
Cellular Structure and Function
Proteome
Transcriptome
Genome
Molecular Scale
Nature Reviews | Molecular Cell Biology
ηm
ηm~μm
μm~mm
mm~cm
1m
1Mm

Fig. 1. Multi-scale characteristics of the biomedical problem space

A study from Harvard School of Public Health estimated that more than 43 million people are injured worldwide each year due to medical errors. These injuries result in the loss of nearly 23 million years of “healthy” life [1]. If we have better information, many of these errors can be avoided. Most of those errors have many secondary victims: the patients’ families, public exchequers, nurses, doctors, social workers, managers, pharmacists involved in the medical care. Most of these errors are not due to any negligence of the physician but are due to the information gaps (imperfect information).

2 The Imperfect Information

Johns Hopkins patient safety experts have analyzed a total of 35,416,020 cases of hospitalizations data. They concluded that medical error is the third leading cause of death in the US following heart disease, and cancer [2]. Most of the medical errors are caused by imperfect or incomplete information. There are different types of medical errors or negative results in healthcare; viz,

1. Wrong Site Surgery (WSS) or “Wrong-site, Wrong-procedure, Wrong-patient errors” (WSPEs) are types of medical errors involves patients who were operated on wrong body part, incorrect procedure, or had a procedure intended for another patient.

2. Another type of medical error is where surgical material like cotton etcetera were left inside the body of the patient.
3. Another type of medical error is misdiagnosis, where doctor in primary care is unable to diagnose the patient correctly. This could be due to symptoms are confusing or overlapping, or the patient failed to communicate with the doctor properly about the disease or it is too early to discover any biomarker. According to a recent study 21% cases where diagnosed wrong; and 66% of patients received a refined or redefined diagnosis [3].
4. There are toxic effects of drug as well. Every medicine is a two-way sword – the same medicine works differently in different bodies. For some patients, the medicine works perfectly as expected, it does not work for some patients; for some patients, it even had toxic effects.
5. Sometime patients do not comply with the medicine dosage or schedules – this causes the drug to become resistant to the disease.
6. If the patient is under multiple medicines at the same time, there are chances of drug-drug interaction with toxic effects.
7. Another area of concern is Hospital acquired infection. Patients in ICU or the Operation Rooms are infected with communicable diseases and some are massively resistant to antibiotics (MRSA infections), and massively on the rise.

Human disease is a complex phenomenon. Combinations of abnormal conditions stemmed from accidental, environmental, lifestyle related, occupational, tobacco, or genetic – make organs or body parts of a person function differently. There are some 10,000 diseases in human body with only about 200 to 300 symptoms or phenotypes. So, there are many overlapping symptoms for multiple diseases. Also, there are many unknown confounding factors that drive a disease. These factors result into confusion and medical errors. When morphologic symptoms are not sufficient to diagnose, the doctor prescribes some pathological tests on body fluids (blood, urine, stool etcetera) to arrive at a diagnosis. Sometime ECG, EEG, Radiology tests are used as well. Recently a new type of test called genetic test is also used where cell DNA are tested to determine the cause of the disease condition. Sometime, there are expensive strategies that can help doctors to make better decisions, but that patients cannot always afford.

In The Law of Medicine book, Dr. Siddhartha Mukherjee, the Pulitzer Prize award winning oncologist wrote, “medicine asks you to make perfect decisions with imperfect information” [4]. This of course was the state of medicine for centuries. With the advent of data sciences and big-data technologies, it is now becoming possible to “help doctors make perfect decisions with perfect information,” we call this ***Evidence Based Precision Medicine***. He also branded medicine as an ***uncertain science***. In pharmacology, randomized clinical trials are in use for a long time. Using mathematical models, it is also becoming possible to bring medicine into mainstream science and predict an outcome or the future onset of a disease or toxicity of a drug. The need of the hour is to increase accuracy and reduce errors to protect both the patient and the physician and restore the dignity of this noble profession. The question we are asking – how to use ***electronic medical records*** (EMR), ***lab*** data, ***clinical*** data, and ***radiology*** data, ***genomic*** data etcetera along with big-data biomedical knowledge in a meaningful way?

3 Biomedical Big-Data

For biomedicine, the scope of big-data is much wider [5]. It involves,

1. Volume (physical volume or the size of the data)
2. Velocity (speed at which an actionable request is serviced)
3. Variety (heterogeneity of the data – multi modal – structured/unstructured)
4. Veracity (security, confidentiality, and reliability)
5. Vexing (algorithmic complexity to process large volume of data)
6. Variability (scale of data – from nano-meters to 1000 s of KM)
7. Value (actionable insight, context based, and functional knowledge)

Every hospital has its private EMR and other patient data in giga-bytes. In addition, there are hundreds of public databases that host biological databases. Many of these databases have already crossed the petabyte (1015) marks [6]. Nucleic Acid Research every year publishes a special issue on Biological databases. This includes some new databases and some updates on the existing databases. The 2017 issue can be found at https://academic.oup.com/nar/article/45/D1/D1/2770636/The-24th-annual-Nucleic-Acids-Research-database. Also, Wikipedia has a page on list of biological databases (https://en.wikipedia.org/wiki/List_of_biological_databases). These lists are only a part of all databases available. The majority of these databases are free and open for public use. These databases include various genomes from human to pathogenic bacteria. They include DNA/RNA sequences, genes, proteins, GO (Gene Ontology), HPO (Human Phenotype Ontology), DO (Disease Ontology), PDB (Protein Data Bank), ChEBI (Chemical Entities of Biological Interest), KEGG (Kyoto Encyclopedia of Genes and Genomes), Drug databases (DrugBank), OMIM (Online Mendelian Inheritance in Man), Recon (Reconstruction of Human Metabolism), dbSNP (DNA Mutation Database), COSMIC (Catalogue of Somatic Mutations in Cancer) etc. There are many databases that host signaling and regulatory networks along with Interactomes. The list goes on and on and on. Today there is a huge body of literature in digital forms like NLM (National Library of Medicine), PubMed, UMLS (Unified Medical Language System), MeSH (Medical Subject Heading), and many more.

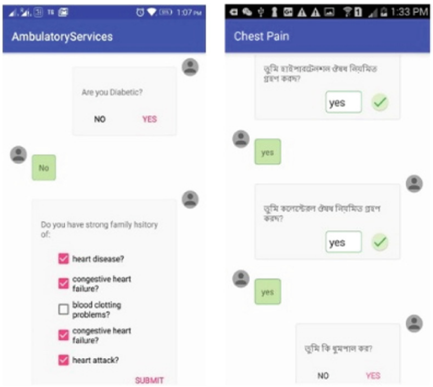
4 NCD and Predictive Medicine

Mortality rates due to infectious or communicable diseases are slowly coming down. Due to prompt actions, a majority of epidemics are contained fast. Communicable diseases are treated in a reactive fashion. However, non-communicable diseases (NCD) follow a different epidemiologic pattern and are on the rise the world over. Lifestyle related diseases, obesity, cardiovascular diseases are becoming chronic. As the average life expectancy increases, cancer and geriatric disease populations are also growing. To manage NCD, reactive medicine needs to become proactive and predictive. Many of them will need to look at predisposition and early onset of diseases. These will need active support from genomic big-data analytics.

5 Outcome Assessment and Chatbot

Outcome of a medical encounter is the efficacy endpoints when developing an intervention for a disease, condition, or procedure. Chatbot is the most effective tool for outcome assessments of a medical encounter because it empowers both patient and the clinician or the proceduralist.

Communicable and non-communicable diseases both have issues with drug compliance. Patients do not always follow the drug dosages and drug schedules. Moreover, doctors often do not receive feedback about either the patient or the effectiveness of a drug. Patients do not always get an opportunity to consult a doctor at a state of emergency. All these gaps and many more can be addressed through smart chatbots driven by AI (Artificial Intelligence). Figure 2 is an example of smart chatbot in English and a local Indian language. This chatbot used ***natural language processor*** (NLP) and ***natural language generator*** (NLG).


AmbulatoryServices

Are you Diabetic?

NO YES

No

Do you have strong family hsitory of:

☑ heart disease?
☑ congestive heart failure?
□ blood clotting problems?
☑ congestive heart failure?
☑ heart attack?

SUBMIT

Chest Pain

তুমি হাইপারটেনশন ঔষধ নিয়মিত গ্রহণ করছ?

yes ✓

yes

তুমি কলেস্টেরল ঔষধ নিয়মিত গ্রহণ করছ?

yes ✓

yes

তুমি কি ধূমপান কর?

NO YES

Fig. 2. AI Driven (NLP & NLG) Chatbots in English and Indian language

6 Health Information Exchange

Until some time ago, hospital electronic medical records (EMR) were paper based. Slowly they are moving towards paper-less systems. To add intelligence and exchange health information between locations, an interoperable system with a centralized exchange and metadata is necessary. Interoperability of spatial and temporal data of a patient information will need data to be normalized across heterogeneous systems and cultures.

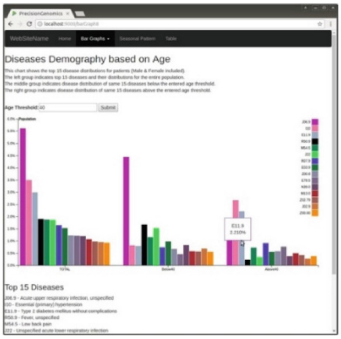
7 A Live Use Case

Here we present the Vibrant Health Analytics Platform System developed by Precision Genomics, the health analytics specialist and sister company of Vibrant Health Limited (http://analytics.vibranthealthsciences.com/). In this analytics system, we have used the glue of ***artificial intelligence*** (AI) and ICD (International Classification of Diseases) to integrate EMR data with Lab data, Radiology data, Genomic data, and many knowledge bases along with many biological databases to show how they should be integrated and analyzed empirically using big-data technology to derive the ***actionable insight*** (AI). Here we have added the artificial intelligence and ***machine learning*** to unleash the medical knowledge from such data. We address the complex challenges of ***data driven medicine***, ***systems biology***, ***genomics***, and ***human diseases*** with their interactions through ***quantitative mathematical models***. We present here a set of ***hypotheses creating systems*** for ***biomedical knowledge discovery*** from clinical data and background knowledge body at the point of care. This knowledge will help reduce the disease burden, increase accuracy, and make healthcare affordable. From the hospital data we created ***data-warehouses*** and ***data-marts*** that were used for the analytics and knowledge discovery using various ***statistical*** and ***quantitative*** models. From these analyses we discover administrative, functional, and biomedical knowledge to be used for ***evidence based precision medicine*** (EBPM).

7.1 Population Health

At the point of care, the doctor needs to study the symptoms or the phenotypes the patient is presented with, then makes a diagnosis followed by some treatment plans. Medical diagnosis is like cracking a one-way cryptographic hash function with a very high collision rate, where diseases have a combination of overlapping output phenotypes. If we know the disease (like a password), we might be able to confirm the symptoms (phenotypes) like whether the password is correct or wrong. But when the patient is presented with some symptoms (or phenotypes), how to reverse-engineer it? This is similar to the case of guessing the correct password quickly. How to arrive at the correct diagnosis of disease and a treatment plan in fastest possible time? Identifying the right chemical (medicine) of the right dosage that will correct the ailment is also a complex process. The root cause of the disease and the statistical impact of a medicine may be obtained from population studies within the hospital or outside.

In an interconnected global network, big-data analytics is the solution. For an effective healthcare, population level knowledge is absolutely necessary. Population health is defined as the health outcomes of a population. Population health also includes the distribution of such health outcomes of groups of individuals within the population. Figure 3 shows the age-wise and sex-wise population health data. It may be noted that the data presented here are live data from a hospital and the diseases presented in this paper used the ICD codes.



Diseases Demography based on Age

This chart shows the top 15 disease distributions for patients (Male & Female included).
The left group indicates top 15 diseases and their distributions for the entire population.
The middle group indicates disease distribution of same 15 diseases below the entered age threshold.
The right group indicates disease distribution of same 15 diseases above the entered age threshold.

Age Threshold: 40

Top 15 Diseases

J06.9 - Acute upper respiratory infection, unspecified
I10 - Essential (primary) hypertension
E11.9 - Type 2 diabetes mellitus without complications
R50.9 - Fever, unspecified
M54.5 - Low back pain
J22 - Unspecified acute lower respiratory infection

Sex-wise Distribution of Diseases

This chart shows the top 10 diseases and their sex-wise distribution. The leftmost group shaows the number of patients in year 2016 having these diseases. The middle group shaows the distribution of these diseases for Male patients; whereas the righmost group shows the distribution of these same top 10 diseases amongst Female patients.

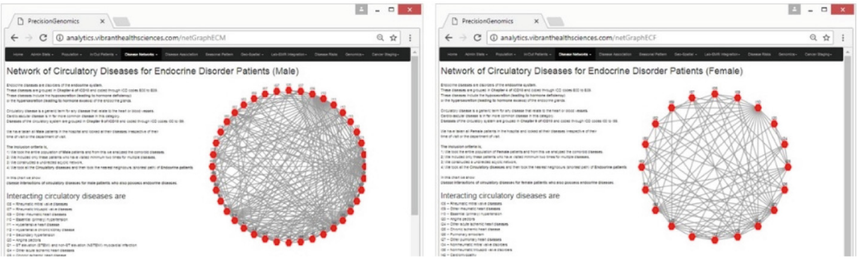
Top 10 Diseases in Male and Female Patients

J06.9 - Acute upper respiratory infection, unspecified
R50.9 - Fever, unspecified
E78.5 - Hyperlipidemia, unspecified
I10 - Essential (primary) hypertension
R05 - Cough
R79.9 - Abnormal finding of blood chemistry, unspecified
E55.9 - Vitamin D deficiency, unspecified
E11.9 - Type 2 diabetes mellitus without complications R10.9 - Unspecified abdominal pain

Fig. 3. Dashboard for population health (disease demography based on age and sex)

7.2 Disease Networks

In the lifetime, a person suffer from many diseases. These diseases have genetic, lifestyle, and environmental roots. In disease network analysis, we have taken these patients who suffer from multiple diseases. We then use these comorbid diseases of all these patients to form an interacting disease network. In Fig. 4 we show the disease relationship network between circulatory disorders (cardiovascular) and endocrine disorders (metabolic and hormone related diseases) of male and female patients in a population. The beauty of this analysis is that this network helps understand the disease-phenotype relationship. It also helps discover colliding phenotypes. In Vibrant Analytics Platform System we have created networks of diseases like neoplasm, metabolic disorders, endocrine diseases, circulatory disorders etcetera.


PrecisionGenomics
analytics.vibranthealthsciences.com/netGraphECM
Home
Disease Networks
PrecisionGenomics
analytics.vibranthealthsciences.com/netGraphECF
Home
Disease Networks

Network of Circulatory Diseases for Endocrine Disorder Patients (Male)

[illegible]

Interacting circulatory diseases are

[illegible]

Network of Circulatory Diseases for Endocrine Disorder Patients (Female)

[illegible]

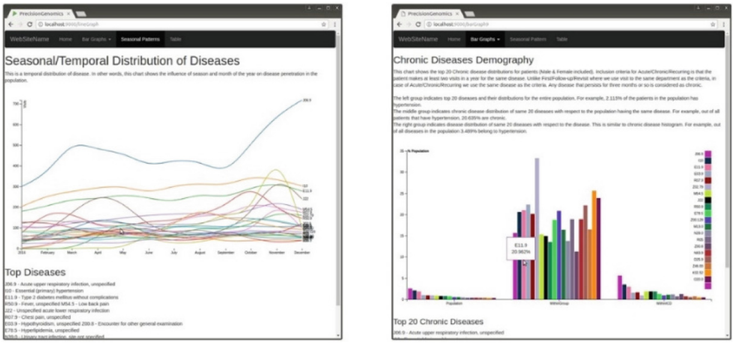
Interacting circulatory diseases are

[illegible]

Fig. 4. Dashboards for disease interactions (circulatory & endocrine disorders)

7.3 Seasonal, Temporal and Chronic Diseases

Seasons have influence on communicable, non-communicable, and chronic diseases. We have models for exhaustive temporal and chronic disease analysis, their characteristics. This analysis is both at patient level (n = 1) and at a population level (n = N). Figure 5 shows the seasonal and chronic disease pattern in the population.



Seasonal/Temporal Distribution of Diseases

This is a temporal distribution of disease. In other words, this chart shows the influence of season and month of the year on disease penetration in the population.

Top Diseases

J06.9 - Acute upper respiratory infection, unspecified
I10 - Essential (primary) hypertension
E11.9 - Type 2 diabetes mellitus without complications
R50.9 - Fever, unspecified M54.5 - Low back pain
J22 - Unspecified acute lower respiratory infection
R07.9 - Chest pain, unspecified
E03.9 - Hypothyroidism, unspecified Z00.8 - Encounter for other general examination
E78.5 - Hyperlipidemia, unspecified
N39.0 - Urinary tract infection, site not specified

Chronic Diseases Demography

This chart shows the top 20 Chronic disease distributions for patients (Male & Female included). Inclusion criteria for Acute/Chronic/Recurring is that the patient makes at least two visits in a year for the same disease. Unlike First/Follow-up/Revisit where we use visit to the same department as the criteria, in case of Acute/Chronic/Recurring we use the same disease as the criteria. Any disease that persists for three months or so is considered as chronic.

The left group indicates top 20 diseases and their distributions for the entire population. For example, 2.115% of the patients in the population has hypertension.
The middle group indicates chronic disease distribution of same 20 diseases with respect to the population having the same disease. For example, out of all patients that have hypertension, 20.635% are chronic.
The right group indicates disease distribution of same 20 diseases with respect to the disease. This is similar to chronic disease histogram. For example, out of all diseases in the population 3.489% belong to hypertension.

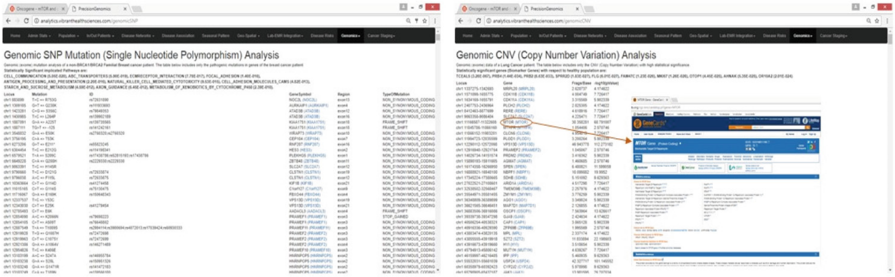
Top 20 Chronic Diseases

J06.9 - Acute upper respiratory infection, unspecified

Fig. 5. The Seasonal and Chronic disease patterns in the population

7.4 Genomics

A person inherits genetic properties and the genetic materials from parents. Sometime children inherit parents' genetic mutations with genetic disease as well at birth (***germline mutations***). Some genetic diseases are sporadic and acquired by patients without any parental origin (***somatic mutations***). These mutations are point-mutations ***single nucleotide polymorphism*** (SNP), ***small insert-deletes***, and ***copy number variation*** (CNV). In SNP mutations a nucleotide in the DNA is mutated causing ***Mendelian disorders***. In CNV however, multiple genes in the chromosomes are deleted or amplified causing complex diseases like cancer. In genomics, DNA is extracted, sequenced using ***next generation sequencers*** (NGS) platforms, which generate ~12 GB of ***exome*** (protein coding genes) unstructured data per patient. Figure 6 shows the genomic analysis of point-mutations, small insert-deletes of a breast cancer patient (left chart) and the CNV analysis of a lung cancer patient (right chart) that were analyzed by Vibrant Analytics Platform System. Vibrant system has identified mTOR (Mechanistic Target Of Rapamycin) gene to be amplified in the lung cancer patient.



Genomic SNP Mutation (Single Nucleotide Polymorphism) Analysis

Genomic (exome) mutation analysis of a non-BRCA1/BRCA2 Familial Breast cancer patient. The table below includes only the pathogenic mutations in genes of the breast cancer patient.

Statistically Significant implicated Pathways are:

Locus | Mutation | ID | GeneSymbol | Region | TypeOfMutation

Genomic CNV (Copy Number Variation) Analysis

Genomic (exome) data of a Lung Cancer patient. The table below includes only the CNV (Copy Number Variation) with high statistical significance

Statistically significant genes (Biomarker Genes) with respect to healthy population are:

Locus | Gene | FragsBase | -log10(pValue)

Fig. 6. Genomic Mutation and CNV analysis for a Breast cancer & Lung cancer patients

7.5 Clinical and Laboratory Data Integration for Predictive Medicine

In this function (Fig. 7) we integrate the Lab data with EMR data along with background knowledge to evaluate the risk factors of an individual. This will help early detection and preventive analysis of various NCD. We also included other calculators like Framingham 10-years cardiovascular disease risk and Cancer staging to arrive at the cancer prognosis. All these models will be used as the ***Doctors' workbench*** at the point of care.

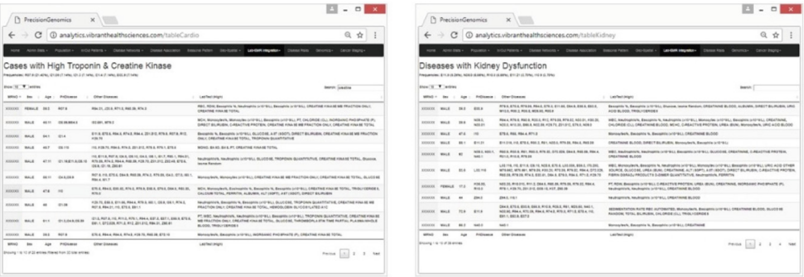

PrecisionGenomics
analytics.vibranthealthsciences.com/tableCardio
Cases with High Troponin & Creatine Kinase
PrecisionGenomics
analytics.vibranthealthsciences.com/tableKidney
Diseases with Kidney Dysfunction

Fig. 7. Risk factors for Cardiovascular and Kidney diseases

7.6 Biostatistics and Epidemiology

In this group, we take biostatistics and epidemiology data and show how they are related within a geographical region (Fig. 8) and used in geo-spatial analysis.

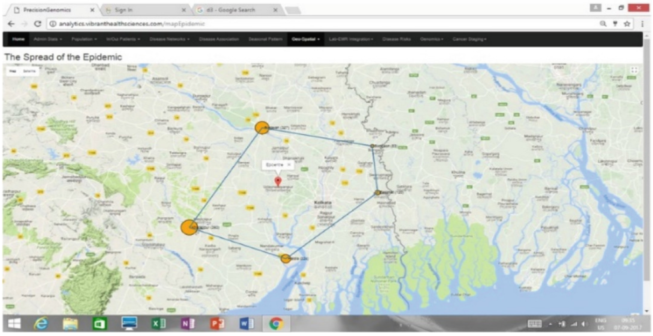

PrecisionGenomics
analytics.vibranthealthsciences.com/mapEpidemic
The Spread of the Epidemic
Kolkata

Fig. 8. The spread of an Epidemic

7.7 KPI Driven Administrative and Operational Knowledge

In this category we analyze the hospital data for operational efficiency. This includes financial, administrative, and operational data like human resources, billing, driven by ***key performance indicators*** (KPI). Figure 9 shows some of the charts like inpatients

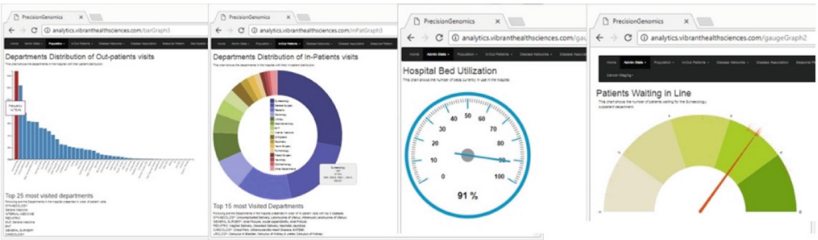

PrecisionGenomics
analytics.vibranthealthsciences.com/barGraph3
Departments Distribution of Out-patients visits
Top 25 most visited departments
PrecisionGenomics
analytics.vibranthealthsciences.com/inPatGraph3
Departments Distribution of In-Patients visits
Top 15 most Visited Departments
PrecisionGenomics
analytics.vibranthealthsciences.com/gau
Hospital Bed Utilization
0
10
20
30
40
50
60
70
80
90
100
91 %
PrecisionGenomics
analytics.vibranthealthsciences.com/gaugeGraph2
Patients Waiting in Line

Fig. 9. Key Performance Indicator (KPI) driven Administrative and Operational Dashboards

and outpatients population, hospital bed utilization, patients standing in line to be examined by the doctor etcetera.

8 Conclusion

Medical data are a combination of structured and unstructured information. Almost the entire disease and clinical information of patients are unstructured data without any parsing rule. Here we presented the Vibrant Analytics Platform System to show how medical data are analyzed from unstructured clinical big-data. We also showed how unstructured genomic big data can be analyzed and presented through dashboards.

We have used many analytics and models to extract actionable insights. These insights are presented through tables, charts, dashboards, and reports. These actionable insights are used to refine the ***clinical pathways***. This insight is then mapped with the spatial and temporal characteristic of the patient at the point of care. This knowledge is also used for population health and training of medical students and nurses. The public health data is used for a better health plan by the regulatory bodies. This knowledge will then be used for ***triple aim*** as proposed by Institute for Health Improvements (http://www.ihi.org/).

Reactive medicine works well with communicable diseases. However, for non-communicable diseases the predictive paradigm needs to be adapted. For this, we have used various models to predict a likely state. We have converted static EMR data using big-data analytics and our integrative methodologies to create actionable medical insights. This information is organized within a knowledgebase structured as a Knowledge repository. The Precision Knowledgebase is a vibrant ontology based probabilistic graph, increasing in intelligence and value exponentially as our information base expands. This dynamic, self-learning Knowledgebase System will be used for ***Evidence Based Precision Medicine*** (EBPM) insights to improve the quality of human health worldwide.

References

1. Jha, A.K., Larizgoitia, I., Audera-Lopez, C., Prasopa-Plaizier, N., Waters, H., Bates, D.W.: The global burden of unsafe medical care: analytic modelling of observational studies. BMJ Qual. Saf. **22**(10), 809–815 (2013)
2. Makary, M.A., Daniel, M.: Medical error—the third leading cause of death in the US. BMJ **353**, i2139 (2016)
3. Van, S.M., Lohr, R., Beckman, T., Naessens, J.M.: Extent of diagnostic agreement among medical referrals. J. Eval. Clin. Prac. **23**, 870–874 (2017)
4. Mukherjee, S.: The Laws of Medicine: Field Notes from an Uncertain Science. Ted Books, New York (2015)
5. Talukder, A.K.: Genomics 3.0: big-data in precision medicine. In: Kumar, N., Bhatnagar, V. (eds.) BDA 2015. LNCS, vol. 9498, pp. 201–215. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-27057-9_14
6. Cook, C.E., et al.: The European bioinformatics institute in 2016: data growth and integration. Nucleic Acids Res. **44**(Database issue), D20–D26 (2016)

Data Mining and Analysis

A Flexible and Efficient Indexing Scheme for Placement of Top-Utility Itemsets for Different Slot Sizes

Parul Chaudhary[1](✉), Anirban Mondal[1](✉),
and Polepalli Krishna Reddy[2](✉)

[1] Shiv Nadar University, Greater Noida, Uttar Pradesh, India
{pc230, anirban.mondal}@snu.edu.in
[2] International Institute of Information Technology,
Hyderabad, Hyderabad, India
pkreddy@iiit.ac.in

Abstract. Utility mining has been emerging as an important area in data mining. While existing works on utility mining have primarily focused on the problem of finding high-utility itemsets from transactional databases, they implicitly assume that each item occupies only one slot. However, in many real-world scenarios, the number of slots consumed by different items typically varies. Hence, this paper considers that a given item may physically occupy any fixed (integer) number of slots. Thus, we address the problem of efficiently determining the top-utility itemsets when a given number of slots is specified as input. The key contributions of our work are three-fold. First, we present an efficient framework to determine the top-utility itemsets for different user-specified number of slots that need to be filled. Second, we propose a novel flexible and efficient index, designated as the STUI index, for facilitating quick retrieval of the top-utility itemsets for a given number of slots. Third, we conducted an extensive performance evaluation using real datasets to demonstrate the overall effectiveness of the proposed indexing scheme in terms of execution time and utility (net revenue) as compared to a recent existing scheme.

Keywords: Utility mining · Index · Top utility itemsets

1 Introduction

The placement of items (products) on the shelf space in retail stores has a significant amount of impact on the sales of the items as well as the revenue generated from sales [1–5]. While the placement of items can be performed manually for relatively small retail stores, it becomes practically infeasible to strategically place items in the shelves of large retail stores in a way that maximizes the sales revenue of the retailer, especially given the *dynamic* nature of the retail environment. Since it is not possible to accurately predict the demand for specific items in advance, decisions concerning the placement of items in the shelves need to be made in a dynamic manner on a near real-time basis depending upon which products are selling out quickly and which products are not so popular with the customers. Intuitively, making such dynamic item placement

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 257–277, 2017.
https://doi.org/10.1007/978-3-319-72413-3_18

decisions for large retail stores on a relatively continuous basis (depending upon customer demand for items) with the objective of maximizing sales revenue constitutes a challenging problem with significant commercial value.

Notably, over the past decade or so, we have been witnessing the popularity and prevalence of mega-sized retail stores with huge retail floor space areas e.g., Walmart Supercenters. Some of the mega-sized retail stores, such as Macy's Department Store at Herald Square (New York City, US) and Shinsegae Centum City Department Store (Busan, South Korea) have more than a million square feet of retail floor space [9]. Furthermore, the largest US retail chains witness about 30% of their annual sales during the Christmas season, and they also see a good percentage of their annual sales during days such as Black Friday [10]. Hence, peak-sales periods typically have a tremendous impact on the annual revenue of the retailers. During these peak-periods, the dynamism in the retail store environment increases dramatically. This occurs primarily because the presence of a huge number of customers and the lack of prior knowledge about their buying preferences for items further exacerbates the challenges associated with predicting the demand for items. Thus, strategic item placement decisions become even more critical for large retailers during peak-sales periods [10].

Consider the case of a large retail store with multiple aisles, where each aisle contains items that are stocked in the slots of the shelves. Observe that some of the slots in the shelves are ***premium slots*** because the items placed in these slots are easily visible as well as physically accessible to the customers. Examples of premium slots include slots that are nearer to the eye level or at the shoulder level of the customers or slots that are at the checkout counters for encouraging impulse buys. Given the relatively higher visibility and physical accessibility to items placed in the premium slots, a given item placed in such slots would have a significantly higher probability of sale than if it were to be placed at other non-premium slots. Non-premium slots are those, which lack visibility and/or physical accessibility. For example, non-premium slots could be located very high or very low in the shelves, thereby making it challenging for customers to even notice items that are placed in these slots. Therefore, when a given item is placed in a non-premium slot, its probability of sale can get significantly reduced. In essence, all the slots in the retail store shelves are not equivalent in terms of the probability of the sale of a given item when placed in a particular slot.

Observe that there would typically be ***multiple blocks*** of premium slots ***of varying sizes*** (in terms of the number of slots) across the different aisles in a given large retail store. Given that the goal of a retail store is to increase its revenue (or profits), it becomes a necessity for the store manager to *strategically* and *dynamically* place the items in premium slots in a way that maximizes revenue. Thus, the store manager needs to decide ***quickly in near real-time*** about which sets of items to place in a large number of premium slots of varying slot sizes (in terms of the number of slots) for maximizing the revenue across the numerous aisles in a large-scale retail store based on customers' near real-time shopping patterns, which can keep changing dynamically. Thus, the ability to make ***highly dynamic*** and ***flexible*** item placement decisions for premium slots is key to maximizing (sales) revenue.

Now if the store manager of a large retail store S were to greedily fill these premium slots with only the most expensive items, the revenue of S would not be maximized and could actually decrease. This is because customers often tend to buy sets of items (i.e., itemsets) together instead of buying just individual items. From the customers' point of view, it is much more convenient for them to have multiple needs met in one location in S instead of walking considerable distances through several different aisles of a large retail store to locate their desired items one-by-one. Alternatively, the store manager of S could be to fill the premium slots with frequent itemsets [6–8]. A frequent itemset refers to the set of items that often appear together in the customer transactions e.g., {bread, jam, milk}. However, existing approaches on frequent itemset mining [6–8] determine frequent itemsets based on the frequency of purchase (support). However, they do not consider item prices (utility values). Observe that prices can vary significantly across items (e.g., the price of a branded suit or a Rolex watch versus the price of a carton of milk or a few loaves of bread). Since revenue depends upon both the frequency of sales and the prices of items, placing frequent itemsets in the premium slots may fail to maximize the revenue of the retailer primarily because some of the frequent itemsets could have low revenue.

Another alternative for the store manager could be to consider a utility mining approach [11–18] towards the problem of deciding upon item placement in the premium slots. Incidentally, utility mining has been emerging as an important area in data mining. The goal of utility mining is to determine high-utility itemsets from transactional databases. Here, utility can be defined in terms of revenue, profits, interestingness and user convenience, depending upon the application. Utility mining approaches focus on creating representations of high-utility itemsets [11], identifying the minimal high-utility itemsets [12], proposing upper-bounds and heuristics for pruning the search space [13, 14] and using specialized data structures, such as the utility-list [15] and the UP-Tree [16], for reducing candidate itemset generation overheads.

These existing works on utility mining cannot address the problem of efficient extraction, indexing and placement of itemsets in premium slots for retail store application scenarios due to three reasons. First, they implicitly assume that a given item occupies only one physical slot in the retail store shelves. However, in real-world retail scenarios, the number of physical slots occupied by different items typically vary. For example, a 500 ml bottle of Pepsi would occupy a considerably lower amount of space in the retail store shelves than over-sized camping equipment. Hence, besides the frequency of sales and the prices of the items, the number of slots occupied by an item also becomes critical for maximizing revenue. Second, existing works are not capable of efficiently indexing and retrieving top-utility itemsets of varying given ***slot sizes***. Here, the slot size of a given item is the number of (integer) slots occupied by that item on the retail store shelves. Third, existing works cannot respond quickly to dynamically changing user shopping patterns on a near real-time basis because they need to first examine all of the candidate high-utility itemsets of different slot sizes before they are able to identify the itemsets of a given slot size.

To address these limitations in existing works, this paper considers that a given item may physically occupy any fixed (integer) number of slots. Thus, we address the problem of efficiently determining the top-utility itemsets when a given number of slots is specified as input. In particular, this paper proposes the STUI indexing scheme, which efficiently determines the top-utility itemsets of any given slot size s. The basic idea of the STUI index is to store only the top-λ high-revenue itemsets for different slot sizes, where each slot size corresponds to a different level of the index. The STUI index uses these top-λ itemsets for progressively building the higher levels of the index one-by-one. Thus, the STUI index is capable of restricting the number of candidate itemsets that need to be examined at each level of the index due to the upper-bound imposed by the value of λ. Furthermore, the proposed STUI index provides ***flexibility*** to the user (e.g., a retail store manager) by serving as a guide towards dynamic and strategic placement of high-utility itemsets in the premium slots.

The key contributions of this work are three-fold:

1. We present an efficient framework to determine the top-utility itemsets for different user-specified number of slots that need to be filled.
2. We propose a novel flexible and efficient index, designated as the STUI index for facilitating quick retrieval of the top-utility itemsets for a given number of slots.
3. We conducted an extensive performance evaluation using real datasets to demonstrate the overall effectiveness of the proposed indexing scheme in terms of execution time and utility (net revenue) as compared to a recent existing scheme.

We shall use revenue as an example of a utility measure throughout this paper; hence, we use the terms *revenue* and *utility* interchangeably. However, our work can also be applied (albeit with minor modifications) to other utility measures as well. Interestingly, although we use retail shelf space as a sample application of our proposed scheme, the problem of item placement in premium slots also has applications in other domains such as Internet advertising. For example, certain webpages (or parts thereof) are more likely to receive eye-ball views and thus contribute to sales. Additionally, note that supply chain negotiations and negotiations of the store manager with the product companies is outside of the scope of this paper. As such, we do not consider these aspects in this paper.

The remainder of this paper is organized as follows. Section 2 discusses related works, while Sect. 3 describes the context of the problem. Section 4 presents the STUI indexing scheme. Section 5 reports the performance evaluation. Finally, we conclude in Sect. 6 with directions for future work.

2 Related Work

Existing works on association rule mining [6–8] consider the problem of determining frequent itemsets based on support. However, they do not take into account the utility values of the items. Moreover, the existing works exploit the downward closure property [6], which implies that the subset of a frequent itemset should also be frequent. However, the downward closure property does not apply to utility mining. Given that the determination of the *optimal* high-utility itemsets would be prohibitively expensive due to the overhead of exhaustive search, it becomes a necessity to design *approximate* approaches for determining high-utility itemsets.

Research efforts on utility mining include the approaches proposed in [11–18]. The work in [16] presents concise representations of high-utility itemsets. Moreover, it discusses the HUG-Miner and GHUI-Miner algorithms to mine those representations. The work in [11] defines the notion of minimal high utility itemsets (MinHUIs) as the smallest itemsets that can generate a large amount of profit. Furthermore, it discusses a representation of minimal high utility itemsets. The work in [12] presents an algorithm, designated as EFIM. EFIM determines high-utility itemsets by using two upper-bounds, namely sub-tree utility and local utility, for pruning the search space. In a similar vein, the proposal in [13] presents an algorithm, designated as EFIM-Closed, for finding closed high-utility itemsets by using pruning strategies in conjunction with upper-bounds for utility. Moreover, the work in [14] prunes the number of candidate itemsets by means of a two-phase algorithm for discovering high-utility itemsets.

The work in [15] mines high-utility itemsets by means of an algorithm, which is designated as the Utility Pattern Growth algorithm for mining. The UP-Growth algorithm uses the Utility Pattern Tree (UP-Tree) for maintaining information about high-utility itemsets. For the generation of candidate itemsets, the UP-Growth algorithm uses pruning strategies. The proposal in [17] discusses an algorithm, designated as the HUI-Miner algorithm, for determining high-utility itemsets. The HUI-Miner algorithm stores utility values and heuristic information about the itemsets in a specialized data structure called the utility-list. In particular, the use of this specialized utility-list data structure facilitates the HUI-Miner algorithm towards avoiding utility computations for a large number of candidate itemsets and also helps the algorithm in avoiding expensive candidate itemset generation. Furthermore, the work in [16] proposes an algorithm, designated as the CHUIMiner algorithm, for computing the utility values of the itemsets without generating candidates. The CHUIMiner algorithm is designed for mining closed high-utility itemsets. The proposal in [18] looks at the problem of determining the top-K high-utility closed patterns from the perspective of business goals. Thus, its goal is to determine the top-K high-utility closed patterns that are related specifically to a given business goal. To achieve this goal, it presents a pruning strategy, whose aim is to prune away low-utility itemsets.

Notably, none of these existing works on utility mining [11–18] consider the notion of the physical space consumed (e.g., the number of slots) by the items. Thus, they do not address the problem of efficiently identifying and retrieving the top-utility itemsets such that the itemsets cover exactly a given number of fixed slots. Moreover, these existing works incur considerable computational costs towards examining a huge number of candidate high-utility itemsets in order to select the top-utility itemsets which would fit into an exact number of given slots. Hence, the approaches proposed in these works are not able to quickly respond to the dynamically changing shopping patterns of the customers, which typically require fast decision-making for placing itemsets for quickly and dynamically filling up the empty premium slots. Thus, these existing works are not capable of effectively maximizing the revenue.

3 Context of the Problem

Consider a finite set Υ of m items $\{i_1, i_2, i_3, .., i_m\}$. Each item of set Υ may be physically different in size. Thus, each item may consume a different number of slots e.g., on the shelves of a retail store. We shall henceforth refer to the number of slots that are physically occupied by a given item as the ***slot size*** of that item. We assume that all slots (premium or non-premium) are of equal size. Each item i_j of set Υ is associated with a price ρ_j, a frequency of sales (support) σ_j and a slot size ω_j. We define the net revenue NR_j of the j^{th} item i_j as the product of its price and support i.e., $NR_j = (\rho_j * \sigma_j)$. Therefore, the net revenue earned per slot is NR_j/ω_j. We shall use the terms revenue and net revenue interchangeably to imply the revenue of any given item or itemset.

This paper addresses the problem of *efficiently* determining the top-λ high-revenue itemsets, given that a user-specified number of slots need to be filled. If we set the value of λ to be high, some of the top-λ itemsets would possibly have low revenue. On the other hand, if the value of λ is set too low, we may miss some itemsets with relatively high revenue. As such, the value of λ is essentially application-dependent; hence, we leave the determination of the optimal value of λ to future work.

Now let us consider the illustrative example in Tables 1 and 2 to better understand the context. Table 1 indicates the respective prices and slot sizes of the items (A to I), while Table 2 depicts a database of five transactions involving these items.

Table 1. Price and slot size information of items

Item	A	B	C	D	E	F	G	H	I
Revenue (ρ)	7	2	6	1	3	1	5	4	3
Slot size (ω)	3	2	1	2	4	3	2	5	2

Definition 1: The net revenue per slot (NR/ω) of a given itemset is computed as the net revenue of the itemset divided by the total number of slots consumed by all the items in the itemset. For example, in Tables 1 and 2, the net revenue of the itemset {A, D} = 48 and the total number of slots consumed by the itemset is (3 + 2) i.e., 5. Therefore, net revenue per slot of itemset {A, D} = 48/5 i.e. 9.6. Similarly, the net revenue of itemset {A, C, G, I} = 63 and the total number of slots consumed by the itemset is 8. Hence, net revenue per slot of itemset {A, C, G, I} = 63/8 i.e. 7.8.

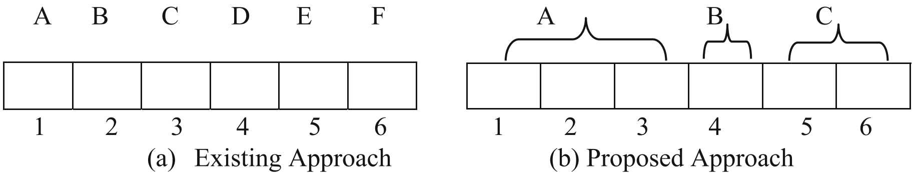
Definition 2: The threshold net revenue per slot ($TH_{NR/\omega}$) of a set S_I of itemsets is computed as $((\mu_{NR/\omega} + (\alpha/100) * \mu_{NR/\omega})$, where $\mu_{NR/\omega}$ is the mean value of NR/ω across all the itemsets in set S_I. Here, α is a parameter which controls the threshold $TH_{NR/\omega}$. The parameter α is application-dependent and its value lies between 0 and 100. The purpose of the parameter α is to act as a lever to limit the number of items satisfying the revenue per slot threshold criterion so that items with low revenue per slot can be effectively pruned away from the index. For example, if the revenue per slot of the items were to follow a uniform distribution, nearly half of the items would satisfy the revenue threshold criterion, if α were set to zero; several of these items could possibly have low revenue. In Table 2, the mean net revenue per slot, $\mu_{NR/\omega} = (9.6 + 4.5 + 6 + 1.84 + 7.8)/5$ i.e., 5.948. Therefore, setting the value of α to 10%, the threshold net revenue per slot ($TH_{NR/\omega}$) = 5.948 + (10/100) * 5.948 i.e., 6.5428.

Definition 3: An ***s*-slot itemset** is defined as an itemset consuming a total of *s* slots i.e., an *s*-slot itemset has a slot size of *s*. In Tables 1 and 2, 5 slots can be filled by the following combinations: {H}, {C, D}, {A, B}, {A, D}, {A, G}, {A, I}, {F, B}, {F, D}, {F, G}, {F, I}, {C, E}. Hence, all these combinations constitute 5-slot itemsets.

Table 2. A sample transactions database

TID	Transaction	Slot (ω)	Frequency of sales (σ)	Net revenue (NR)	NR/ω
1	A, D	5	6	48	9.6
2	B, C, I, F	8	3	36	4.5
3	A, C, G	6	2	36	6
4	A, B, C, G, H	13	1	24	1.84
5	A, C, G, I	8	3	63	7.8

Figure 1(a) depicts the underlying assumption of the existing approach, where each of the items (A to E) physically consumes only one slot. Figure 1(b) indicates the assumption of the proposed approach, where the number of physical slots consumed by the items may vary. For example item A consumes 3 slots, item B consumes 1 slot and item C consumes 2 slots. Observe how the assumption of the proposed approach is consistent with real-world scenarios, where the physical sizes of items typically vary e.g., Pepsi or ketchup bottle versus large-sized camping equipment.



(a) Existing Approach (b) Proposed Approach

Fig. 1. Underlying assumptions of the existing approach and the proposed approach

Intuitively, given a large number of items in set ϒ, the number of possible combinations of the items satisfying a given slot size would essentially explode. Hence, a naïve brute-force approach of generating and examining all possible itemsets of a given slot size for finding the top-λ high-revenue itemsets would be prohibitively expensive. Hence, it becomes imperative to set threshold values for the price, support and slot size of the items in set ϒ as well as the itemsets of different slot sizes arising from set ϒ. We defer the determination of these threshold values to the next section.

4 Proposed Scheme

This section first discusses the basic idea of the STUI index and then describes the index. Moreover, we present an example and the algorithm for building the index.

4.1 Basic Idea

Our proposed STUI indexing scheme aims at efficiently determining the top-utility itemsets of any given slot size, given that the individual slot sizes of the items may vary. The basic idea of the STUI index is to store only the top-λ high-revenue itemsets for each different slot size instead of storing all the itemsets of a given slot size. The s^{th} level of the STUI index corresponds to itemsets having a slot size of s. The index is built in a level-wise manner starting from the lowest level, which corresponds to itemsets of slot size 1. Then the next higher levels of the index are built progressively one-by-one by considering only the top-λ high-revenue itemsets at the lower levels.

By maintaining only the top-λ itemsets, the STUI index restricts the number of candidate itemsets that need to be examined for building the next higher level of the index. This improves the efficiency of computation for building the index because a lower number of candidate itemsets need to be examined. This also reduces index storage costs since the number of itemsets being maintained at each level of the index is upper-limited by the value of λ. Furthermore, the STUI index also facilitates quick retrieval of high-revenue itemsets of any given slot size. This is because the top high-revenue itemsets are maintained in the index for different slot sizes.

4.2 Description of the STUI Index

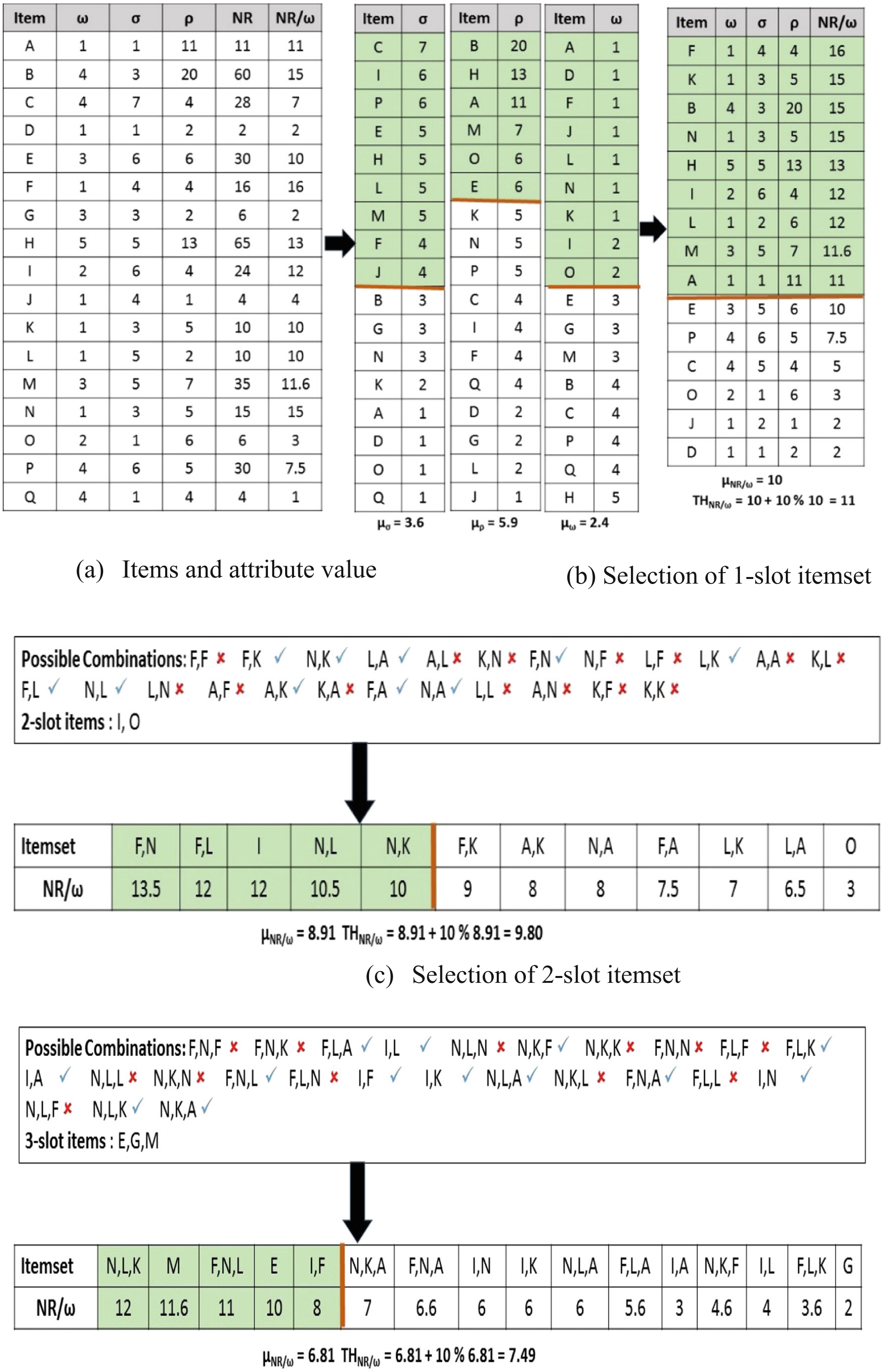
The STUI index is essentially a multi-level index, where each level corresponds to a given slot size s. Corresponding to each level, the STUI index stores the top-λ high-revenue itemsets of the slot size associated with that level.

Each level in the STUI index corresponds to a hash bucket. Thus, for indexing itemsets of N different slot sizes, the index would contain N hash buckets i.e., one hash bucket per slot size. Hence, when a query Q tries to find the top-utility itemsets of a given slot size s, Q is able to traverse quickly to the s^{th} hash bucket as opposed to having to traverse through all the hash buckets corresponding to $s = \{1, 2, \ldots, s - 1\}$.

Now, for each level i in the STUI index, the corresponding hash bucket contains a pointer to a linked list of the top-λ high-revenue itemsets of slot size s. The entries of the linked list are of the form (*itemset*, σ, ρ, NR/ω), where *itemset* refers to the given itemset under consideration. Here, σ is the support (frequency of sales) of *itemset*, while ρ refers to the total price of all the items in *itemset.* NR/ω is the net revenue per slot, as discussed earlier in Sect. 3 (see Definition 1). The entries in the linked list are sorted in descending order of their values of NR/ω.

4.3 Illustrative Example for Building the STUI Index

Figure 2 depicts an illustrative example for the creation of the STUI index. Figure 2(a) indicates 17 items, namely A to Q, with their respective values of slot size ω, support σ, price ρ, net revenue NR and net revenue per slot (NR/ω). For example, in Fig. 2(a), the values of ω, σ, ρ, NR and NR/ω for items B and G are {4, 3, 20, 60, 15} and {3, 3, 2, 6, 2} respectively. For simplicity, we set λ = 5 in this example i.e., for each slot size, there would be at most 5 top-revenue per slot itemsets.



Item	ω	σ	ρ	NR	NR/ω
A	1	1	11	11	11
B	4	3	20	60	15
C	4	7	4	28	7
D	1	1	2	2	2
E	3	6	6	30	10
F	1	4	4	16	16
G	3	3	2	6	2
H	5	5	13	65	13
I	2	6	4	24	12
J	1	4	1	4	4
K	1	3	5	10	10
L	1	5	2	10	10
M	3	5	7	35	11.6
N	1	3	5	15	15
O	2	1	6	6	3
P	4	6	5	30	7.5
Q	4	1	4	4	1

(a) Items and attribute value

Item	σ
C	7
I	6
P	6
E	5
H	5
L	5
M	5
F	4
J	4
B	3
G	3
N	3
K	2
A	1
D	1
O	1
Q	1

$\mu_{\sigma} = 3.6$

Item	ρ
B	20
H	13
A	11
M	7
O	6
E	6
K	5
N	5
P	5
C	4
I	4
F	4
Q	4
D	2
G	2
L	2
J	1

$\mu_{\rho} = 5.9$

Item	ω
A	1
D	1
F	1
J	1
L	1
N	1
K	1
I	2
O	2
E	3
G	3
M	3
B	4
C	4
P	4
Q	4
H	5

$\mu_{\omega} = 2.4$

Item	ω	σ	ρ	NR/ω
F	1	4	4	16
K	1	3	5	15
B	4	3	20	15
N	1	3	5	15
H	5	5	13	13
I	2	6	4	12
L	1	2	6	12
M	3	5	7	11.6
A	1	1	11	11
E	3	5	6	10
P	4	6	5	7.5
C	4	5	4	5
O	2	1	6	3
J	1	2	1	2
D	1	1	2	2

$\mu_{NR/\omega} = 10$

$TH_{NR/\omega} = 10 + 10\,\%\,10 = 11$

(b) Selection of 1-slot itemset

Possible Combinations: F,F ✗ F,K ✓ N,K ✓ L,A ✓ A,L ✗ K,N ✗ F,N ✓ N,F ✗ L,F ✗ L,K ✓ A,A ✗ K,L ✗ F,L ✓ N,L ✓ L,N ✗ A,F ✗ A,K ✓ K,A ✗ F,A ✓ N,A ✓ L,L ✗ A,N ✗ K,F ✗ K,K ✗

2-slot items : I, O

Itemset	F,N	F,L	I	N,L	N,K	F,K	A,K	N,A	F,A	L,K	L,A	O
NR/ω	13.5	12	12	10.5	10	9	8	8	7.5	7	6.5	3

$\mu_{NR/\omega} = 8.91$ $TH_{NR/\omega} = 8.91 + 10\,\%\,8.91 = 9.80$

(c) Selection of 2-slot itemset

Possible Combinations: F,N,F ✗ F,N,K ✗ F,L,A ✓ I,L ✓ N,L,N ✗ N,K,F ✓ N,K,K ✗ F,N,N ✗ F,L,F ✗ F,L,K ✓ I,A ✓ N,L,L ✗ N,K,N ✗ F,N,L ✓ F,L,N ✗ I,F ✓ I,K ✓ N,L,A ✓ N,K,L ✗ F,N,A ✓ F,L,L ✗ I,N ✓ N,L,F ✗ N,L,K ✓ N,K,A ✓

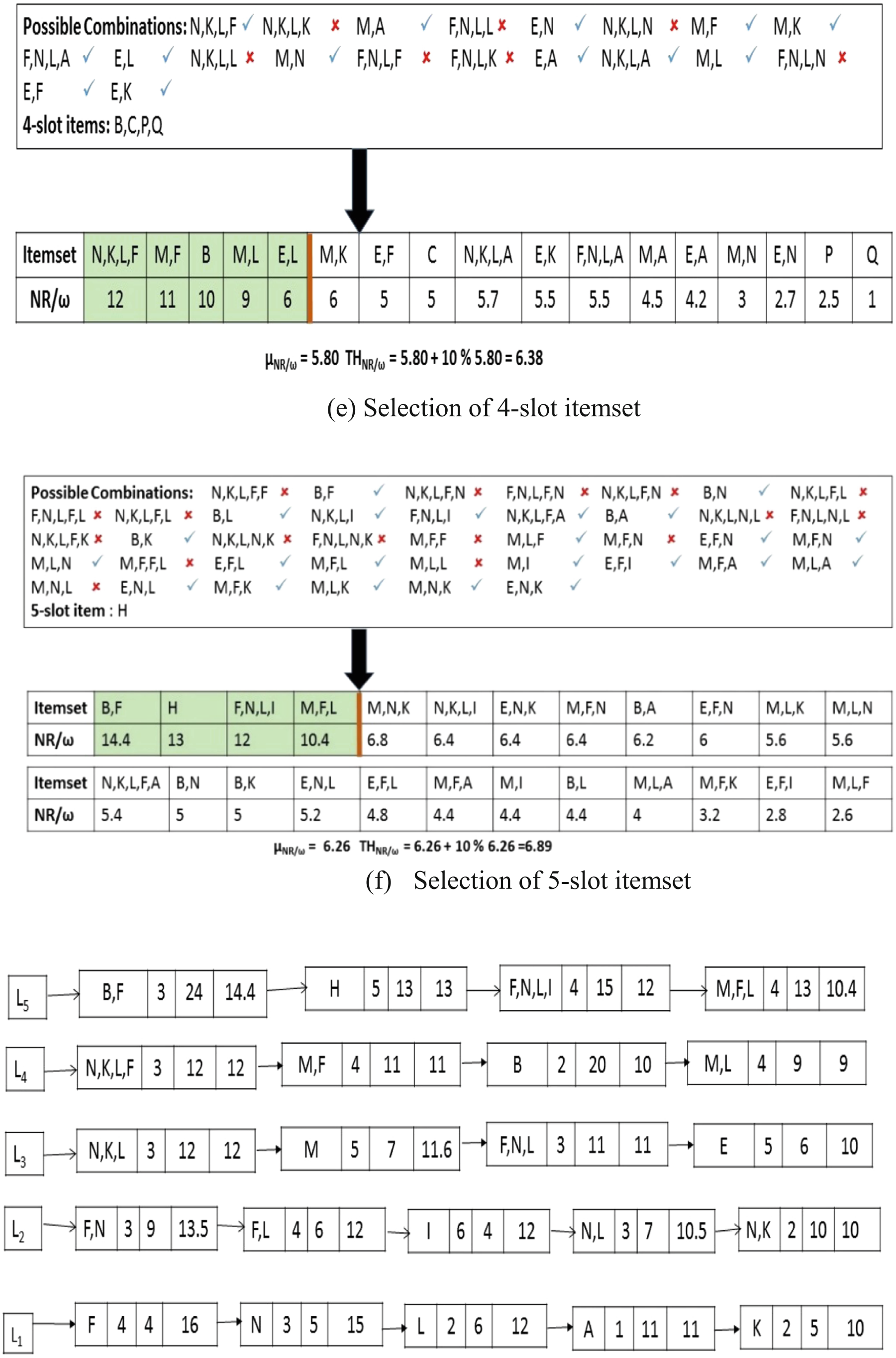
3-slot items : E,G,M

Itemset	N,L,K	M	F,N,L	E	I,F	N,K,A	F,N,A	I,N	I,K	N,L,A	F,L,A	I,A	N,K,F	I,L	F,L,K	G
NR/ω	12	11.6	11	10	8	7	6.6	6	6	6	5.6	3	4.6	4	3.6	2

$\mu_{NR/\omega} = 6.81$ $TH_{NR/\omega} = 6.81 + 10\,\%\,6.81 = 7.49$

(d) Selection of 3-slot itemset

Fig. 2. Illustrative example for building the STUI Index (Color figure online)



Possible Combinations: N,K,L,F ✓ N,K,L,K ✗ M,A ✓ F,N,L,L ✗ E,N ✓ N,K,L,N ✗ M,F ✓ M,K ✓
F,N,L,A ✓ E,L ✓ N,K,L,L ✗ M,N ✓ F,N,L,F ✗ F,N,L,K ✗ E,A ✓ N,K,L,A ✓ M,L ✓ F,N,L,N ✗
E,F ✓ E,K ✓
4-slot items: B,C,P,Q

Itemset	N,K,L,F	M,F	B	M,L	E,L	M,K	E,F	C	N,K,L,A	E,K	F,N,L,A	M,A	E,A	M,N	E,N	P	Q
NR/ω	12	11	10	9	6	6	5	5	5.7	5.5	5.5	4.5	4.2	3	2.7	2.5	1

$\mu_{NR/\omega}$ = 5.80 $TH_{NR/\omega}$ = 5.80 + 10 % 5.80 = 6.38

(e) Selection of 4-slot itemset

Possible Combinations: N,K,L,F,F ✗ B,F ✓ N,K,L,F,N ✗ F,N,L,F,N ✗ N,K,L,F,N ✗ B,N ✓ N,K,L,F,L ✗
F,N,L,F,L ✗ N,K,L,F,L ✗ B,L ✓ N,K,L,I ✓ F,N,L,I ✓ N,K,L,F,A ✓ B,A ✓ N,K,L,N,L ✗ F,N,L,N,L ✗
N,K,L,F,K ✗ B,K ✓ N,K,L,N,K ✗ F,N,L,N,K ✗ M,F,F ✗ M,L,F ✓ M,F,N ✗ E,F,N ✓ M,F,N ✓
M,L,N ✓ M,F,F,L ✗ E,F,L ✓ M,F,L ✓ M,L,L ✗ M,I ✓ E,F,I ✓ M,F,A ✓ M,L,A ✓
M,N,L ✗ E,N,L ✓ M,F,K ✓ M,L,K ✓ M,N,K ✓ E,N,K ✓
5-slot item : H

Itemset	B,F	H	F,N,L,I	M,F,L	M,N,K	N,K,L,I	E,N,K	M,F,N	B,A	E,F,N	M,L,K	M,L,N
NR/ω	14.4	13	12	10.4	6.8	6.4	6.4	6.4	6.2	6	5.6	5.6

Itemset	N,K,L,F,A	B,N	B,K	E,N,L	E,F,L	M,F,A	M,I	B,L	M,L,A	M,F,K	E,F,I	M,L,F
NR/ω	5.4	5	5	5.2	4.8	4.4	4.4	4.4	4	3.2	2.8	2.6

$\mu_{NR/\omega}$ = 6.26 $TH_{NR/\omega}$ = 6.26 + 10 % 6.26 =6.89

(f) Selection of 5-slot itemset

L_5 → B,F | 3 | 24 | 14.4 → H | 5 | 13 | 13 → F,N,L,I | 4 | 15 | 12 → M,F,L | 4 | 13 | 10.4

L_4 → N,K,L,F | 3 | 12 | 12 → M,F | 4 | 11 | 11 → B | 2 | 20 | 10 → M,L | 4 | 9 | 9

L_3 → N,K,L | 3 | 12 | 12 → M | 5 | 7 | 11.6 → F,N,L | 3 | 11 | 11 → E | 5 | 6 | 10

L_2 → F,N | 3 | 9 | 13.5 → F,L | 4 | 6 | 12 → I | 6 | 4 | 12 → N,L | 3 | 7 | 10.5 → N,K | 2 | 10 | 10

L_1 → F | 4 | 4 | 16 → N | 3 | 5 | 15 → L | 2 | 6 | 12 → A | 1 | 11 | 11 → K | 2 | 5 | 10

(g) The corresponding STUI index

Fig. 2. (*continued*)

Observe how in Fig. 2(b), the items are sorted separately thrice i.e., once in descending order of their respective support values, once in terms of their prices, and once in ascending order of their slot sizes. The rationale here is that we want to consider items that have either high support or high price or low value of slot size because these are the items that may have relatively higher net revenue per slot. For example, in Fig. 2(b), since the mean support $\mu_\sigma = 3.6$, only the items {C, I, P, E, H, L, M, F, J} are selected for having support values equal to or exceeding the value of μ_σ. Similarly, since the mean price $\mu_\rho = 5.9$, only the items {B, H, A, M, O, E} are selected. The selected items are shaded in green for easy readability. Observe how the selected items are combined using the set union operation to remove duplicates. Since the revenue threshold value $TH_{NR/\omega} = 11$, only the items {F, B, N, H, I, L, M, A, K} are selected. Since $\lambda = 5$ in this example, only the top-5 of these items (in terms of NR/ω) are inserted into level L_1 of the index.

Figure 2(c) indicates all the possible combinations of itemsets of slot size 2 based on the items in level L_1 of the index and the items with slot size of 2. Since the ordering of the items in a given itemset does not matter, we remove the duplicates. The itemsets marked with a tick symbol are the candidate itemsets, while the itemsets marked with the cross symbol are duplicates, hence they are not candidate itemsets. For example, {F, N} is marked with a tick symbol, hence it is a candidate itemset. However, {N, F} is marked with a cross symbol, hence it is not a candidate itemset. In Fig. 2(c), observe how only the itemsets with revenue per slot equal to or above the value of TH_{NR} (i.e., 9.80) are selected to be inserted into level L_2 of the index.

In Fig. 2(d), notice how the different combinations of itemsets of slot size 3 are created by combining the itemsets from L_1 and L_2 of the index. Since itemsets, such as {F, N, F} and {F, N, N} are itemsets of slot size 2, they are not considered as candidate itemsets for building the level L_2 of the index; hence, they are marked with a cross symbol. Similarly, Fig. 2(e) indicates how the itemsets of slot size 4 are selected into level L_4 of the index. Figure 2(f) indicates how the itemsets of slot size 5 are selected into level L_5 of the index. Finally, Fig. 2(g) depicts the corresponding STUI index.

4.4 Creation of the STUI Index

Using the intuition from the illustrative example in Fig. 2, we will now discuss the creation of the STUI index. First, for slot size of 1, we select only those items, which have a slot size of 1 and whose net revenue per slot is either equal to or above a given threshold value. Notably, the purpose of this threshold value is to ensure that items with low revenue per slot (or itemsets, in case of higher slot sizes) are efficiently pruned away from the index. Then we sort the selected items in descending order of their values of revenue per slot and insert the top-λ items into level 1 of the index. Next, from level 1 of the index, we list all the combinations of the items of slot size 1 and we also list the individual items with slot size of 2. Among these items/itemsets, we select only those, whose revenue per slot is equal to or exceeds a specific revenue per slot threshold due to the rationale explained earlier. Among these itemsets, the top-λ high-revenue itemsets are now inserted into level 2 of the index.

Then, for creating itemsets of slot size 3, we list all of the possible combinations of the items in level 1 of the STUI index and the itemsets in level 2 of the index. Additionally, individual items with slot size of 3 are also listed. Among these itemsets of slot size 3, we select only the top-λ high-revenue itemsets, whose revenue either equals or exceeds a given threshold; these selected itemsets are then inserted into level 3 of the index. This process is continued till the maximum level of the index has been populated with the entries of itemsets corresponding to the slot size at that level.

Now let us make a few important observations about the STUI index creation algorithm. First, when we build the higher levels of the index, only the top-λ high-revenue itemsets in the lower levels of the index are considered, thereby implicitly restricting the number of candidate itemsets corresponding to each different slot size. Intuitively, we can understand that this prevents the explosion in the total number of itemsets that need to be examined for building the next higher levels of the index. Second, for creating higher levels of the STUI index, we need to examine multiple lower levels of the index. For example, for building level 6 of the index (i.e., the level of the index containing itemsets with a slot size of 6), we need to combine itemsets of slot sizes 1 and 5, as well as itemsets of slot sizes 2 and 4. Similarly, to build level 9 of the index, we need to combine itemsets of the following slot sizes: (1, 8), (2, 7), (3, 6) and (4, 5).

Algorithm 1 depicts the algorithm for the creation of the STUI index. Lines 1–14 indicate the building of the first level L_1 of the index i.e., for itemsets of slot size 1. Observe how the items are sorted thrice separately in terms of support, price and slot size. Only those items with high support or high price or low slot size qualify as candidates for the first level of the index. Notably, the rationale for this is to ensure that these candidate items have relatively high net revenue per slot. Among these candidates, the top-λ itemsets in terms of net revenue per slot are inserted into the first level L_1 of the index. Moreover, from Line 10, observe how the duplicate items are removed by performing the union operation on sets A, B and C. Furthermore, in Line 12, the value of $TH_{NR/\omega}$ is computed as discussed in Sect. 3 (see Definition 2).

Algorithm 1. Algorithm for creating the STUI index

```
Inputs: a) Set $\Upsilon$ of items, where $\omega_j$, $\sigma_j$, $\rho_j$, $NR_j$ and $NR/\omega_j$ represent the respective slot,
          support, price, net revenue and net revenue per slot of the j$^{th}$ item
        b) N = user-specified maximum level of the index
Output: STUI Index
Begin
/* Building level 1 of the index */
1. Sort the items in set $\Upsilon$ in descending order of $\sigma$ into list $\Upsilon'$
2. Compute $\mu_\sigma$ for the items in list $\Upsilon'$/* mean support value */
3. Select items with value of $\sigma$ above or equal to $\mu_\sigma$ into Set A
4. Sort the items in set $\Upsilon$ in descending order of $\rho$ into list $\Upsilon''$
5. Compute $\mu_\rho$ for the items in list $\Upsilon''$ /* mean price value */
6. Select items with value of $\rho$ above or equal to $\mu_\rho$ into Set B
7. Sort the items in set $\Upsilon$ in ascending order of $\omega$ into list $\Upsilon'''$
8. Compute $\mu_\omega$ for the items in list $\Upsilon'''$ /* mean slot size value */
9. Select items with value of $\omega$ below or equal to $\mu_\omega$ into Set C
10. Compute set X = A U B U C /* union of sets A, B and C */
11. Sort the items in set X in descending order of net revenue per slot into list X'
12. Compute the value of $TH_{NR/\omega}$ for the items in list X'
13. From list X', select the top-$\lambda$ items whose net revenue per slot is equal to or above
    the value of $TH_{NR/\omega}$
14. Insert these top-$\lambda$ items into level $L_1$ of the index

/* Building the intermediate levels of the index one-by-one */
15. for (i = 2 to N) /* N is the maximum level of the index */
16.     Create combinations of itemsets of slot size i from the index entries in $L_1$ till level
        $L_{i-1}$ of the index into list Y such that the sum of the index levels selected for the
        combination of itemsets must be equal to i
17.     Remove duplicate itemsets from list Y
18.     Select items with the value of $\omega$ equal to i from set $\Upsilon$ and add to list Y
19.     Sort itemsets in list Y in descending order of net revenue per slot
20.     Compute the value of $TH_{NR/\omega}$ for the itemsets in list Y
21.     From list Y, select the top-$\lambda$ itemsets whose net revenue per slot is equal to or
        above the value of $TH_{NR/\omega}$
22.     Insert these top-$\lambda$ itemsets into level $L_i$ of the index
End
```

Lines 15–22 indicate how the intermediate levels (i.e., level 2 to the maximum level N) of the STUI index are built one-by-one. In Line 16, observe how the ith level of the index is created by examining all the possible combinations of itemsets from level 1 till level $(i - 1)$ of the index such that the sum of the selected index levels selected must be equal to i. For example, for building the level 5 (i.e., L_5) of the index, we will consider itemsets from level L_1 till $L_{4.}$ In this case, the combinations of index levels selected for building the level 5 of the index are (L_1, L_4) and (L_2, L_3).

5 Performance Evaluation

This section reports the performance evaluation by comparing the proposed STUI index w.r.t a recent existing scheme [12]. We have implemented both the proposed scheme as well as the reference scheme in Java. All of our experiments were performed on a 64 bit Core i5 processor running Windows 7 with 8 GB memory.

For our experiments, we used two real-world datasets, namely *Retail* and *Connect*. These datasets were obtained from the SPMF open-source data mining library [19]. Table 3 summarizes the number of items and the number of transactions associated with each of these datasets. Incidentally, the Retail and Connect datasets do not provide utility values. Hence, we generated the utility values of the items to be in the range of \$1 to \$100 as follows. We divided the utility value range into three buckets, namely, *low*, *mid* and *high*. The *low* bucket corresponds to \$1–\$10, and 30% of the total items are randomly assigned to this bucket. The *mid* and *high* buckets correspond to \$11–\$75 and \$76–\$100 respectively, and 60% and 10% of the total items are randomly assigned respectively to these buckets. Now given a specific item, we first identify its assigned bucket. Then its utility value is assigned randomly between the lower and upper range of utility values corresponding to that bucket.

Table 3. Statistical information about datasets

Dataset	No. of items	No. of transactions
Retail	16,470	88,162
Connect	129	67,557

Notably, the physical space consumed by the items can vary in terms of the slot size (i.e., the number of slots) occupied by a given item. Each item in the dataset is randomly assigned a value of slot size between 1 and 10 as follows. The slot size range is divided into three buckets, namely, *small*, *medium* and *large*. The *small* bucket corresponds to slot sizes between 1 and 3; the *medium* bucket corresponds to slot sizes between 4 and 7 and the *large* bucket corresponds to slot sizes between 8 and 10. Given a specific item, we identify its assigned bucket and then randomly assign it a value of slot size such that the value is within the respective lower range and the upper range of the slot sizes corresponding to that bucket.

The parameters of our simulation were selected to closely reflect real-world scenarios based on our understanding of the application environment. Moreover, the parameters of our simulation are based on existing works [12, 16, 20]. Table 4 summarizes the parameters of our performance evaluation. From Table 4, observe that we set the parameter α, which controls the revenue threshold, to 30% for all of our experiments. We set the number λ of top high-utility items per level of the index to 20

Table 4. Parameters for the performance evaluation

Parameter	Default	Variations
Revenue threshold control parameter (α)	30%	
Top high-utility items (λ)	20	40, 60, 80, 100
Queried number of Slots (*s*)	4	2, 6, 8, 10

as the default. We also vary the value of λ to study its impact on the performance. Furthermore, we set the queried itemset slot size *s* to 4 as the default. We also vary the value of *s* in our experiments to study the impact on the performance of the schemes.

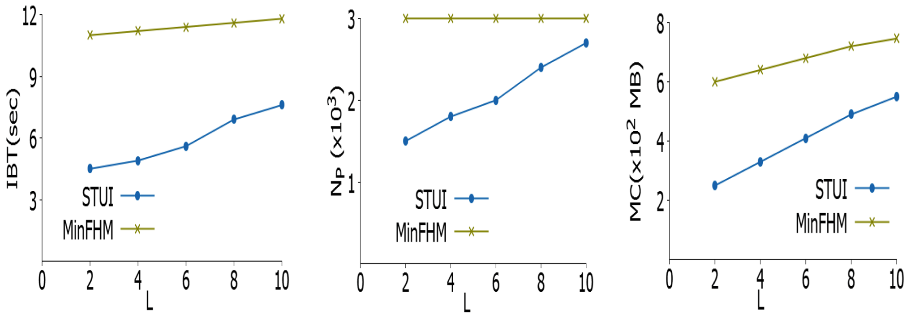
For performance comparison purposes, we adapted the recent MinFHM scheme [12] as follows. First, we use the MinFHM scheme to generate all of the itemsets consuming different slot sizes. Second, from these generated itemsets, we extracted all of the itemsets corresponding to each slot size. For example, we extracted the itemsets corresponding to slot size $s = 1$; then we extracted the itemsets corresponding to slot size $s = 2$ and so on all the way up to the pre-defined maximum slot size supported by our system. Third, from these extracted itemsets of the given slot size *s*, we randomly selected any λ itemsets as the query result. We shall henceforth refer to this scheme as **MinFHM**.

The performance metrics are index build time (IBT), execution time (ET), memory consumption (MC), number of patterns (N_P) and net revenue (NR). IBT is the time required for building the index. ET is the average execution time of a query for determining the top-λ itemsets of any given user-specified slot size *s*. $ET = \frac{1}{N_C}\sum_{q=1}^{N_C}\left(t_f - t_o\right)$, where t_o is the query-issuing time, t_f is the time of the query result reaching the query-issuer, and N_C is the total number of the queries. MC is the total memory consumption of a given scheme for building its index. N_P is the number of patterns (itemsets) that a given scheme needs to examine for answering a specific query. Given a query, the query result comprises λ itemsets. NR is the total revenue of all these λ itemsets. Thus, $NR = \sum_{j=1}^{\lambda} R_j$, where R_j is the revenue of the j^{th} itemset.

5.1 Performance of Index Creation

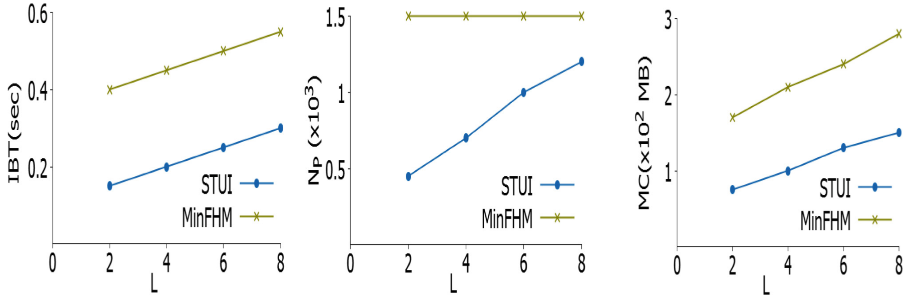
We performed an experiment to study the performance of index creation. Figures 3 and 4 depict the results for the Retail and Connect datasets respectively. The results in Fig. 3(a) indicate that STUI incurs a considerably lower value of IBT (index build time) than MinFHM. This occurs because unlike MinFHM, STUI considers only the top-λ itemsets of a given slot size for building the index at each level. Thus, in contrast with MinFHM, STUI significantly restricts the number of itemsets that need to be examined at each level for building the index.

On the other hand, in case of MinFHM, IBT remains comparable across different values of L (the number of levels in the index). This is because MinFHM incurs its predominant cost in generating all of the itemsets of different slot sizes before building its index. After all of the itemsets have been generated, segregating the itemsets across the different levels of the index based on slot size constitutes a relatively minor overhead. Hence, as the results in Fig. 3(b) indicate, the number N_P of patterns, which



(a) Index Build Time (b) Number of Patterns (c) Memory Consumption

Fig. 3. Performance of index creation (Retail dataset)



(a) Index Build Time (b) Number of Patterns (c) Memory Consumption

Fig. 4. Performance of index creation (Connect dataset)

need to be examined for building the index in case of MinFHM, remains comparable across the different values of L.

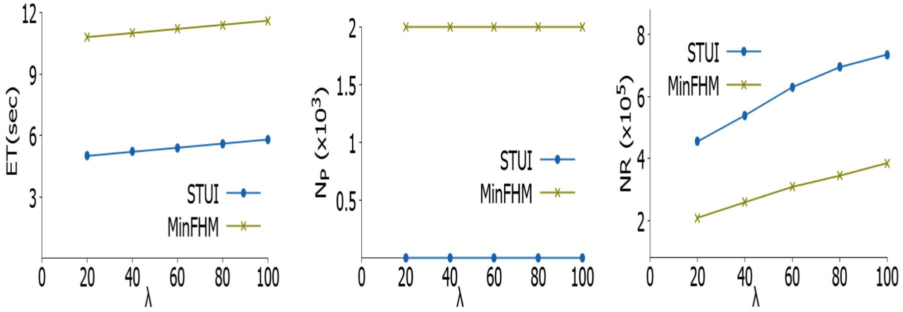
In contrast, STUI builds its index in a level-by-level manner. Hence, its index build time increases as the number L of levels in the index increases. Furthermore, as the value of L increases, a considerably higher number of itemset combinations qualify w.r.t. the revenue per space threshold criterion of STUI. This occurs because for building each level of the index, STUI considers itemset combinations from multiple lower levels of the index e.g. for building level 6, itemset combinations are formed from levels 1 and 5, as well as from levels 2 and 4. Hence, as the results in Fig. 3(b) indicate, the number N_P of patterns, which need to be examined, for building different levels of the STUI index increase with increase in the value of L.

The results in Fig. 3(c) indicate that both of the schemes incur higher memory consumption (MC) as the value of L increases. This occurs because an increased number of levels in the index results in more storage requirements for maintaining the top-λ itemsets at each respective level. STUI outperforms MinFHM in terms of MC because the value of λ and the revenue threshold of STUI significantly limits the generation of the candidate itemsets at each level of the index. The results in Fig. 4

follow similar trends as those of Fig. 3; the actual values are lower in case of the results in Fig. 4 since the Connect dataset has a significantly lower number of items as well as a lower number of transactions as compared to those of the Retail dataset.

5.2 Effect of Variations in λ

Figures 5 and 6 depict the effect of variations in λ for the Retail dataset and the Connect dataset respectively. The results in Figs. 5(a) and (b) indicate that STUI incurs significantly lower execution time (ET) and it has to examine a considerably lower number of patterns (itemsets) as compared to those of MinFHM. This occurs because when a query comes in to determine the top-λ itemsets of slot size *s*, STUI just needs to traverse to the level in the index corresponding to *s*. Then it simply needs to retrieve the λ itemsets from the linked list entries at that level of the index. In contrast, MinFHM needs to first generate all of the itemsets of the queried slot size *s*. Then it needs to extract all the itemsets of slot size *s*. Finally, it has to randomly select any λ itemsets from these extracted itemsets as the query result.

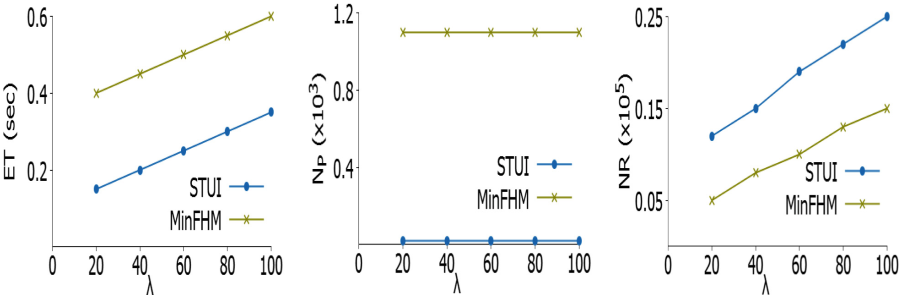


(a) Execution Time (b) Number of Patterns (c) Net Revenue

Fig. 5. Effect of variations in λ (Retail dataset)

As the value of λ increases, ET remains comparable for MinFHM since it incurs its predominant cost in generating all the itemsets of varied slot sizes and then extracting the itemsets of the queried slot sizes. Consequently, the random selection of any λ itemsets of the queried slot size from the extracted itemsets represents a relatively minor overhead. On the other hand, as the value of λ increases, ET increases slightly for STUI because STUI needs to traverse more linked list entries to retrieve more of the top-λ itemsets in response to increased values of λ.

The results in Fig. 5(c) indicate that both schemes exhibit higher values of net revenue (NR) with increase in the value of λ. This occurs because as the value of λ increases, more itemsets are retrieved as the query result for both of the schemes; an increased number of retrieved itemsets imply more net revenue. STUI shows significantly higher values of NR as compared to that of MinFHM because MinFHM randomly selects the λ itemsets, while STUI is able to directly select the top-λ



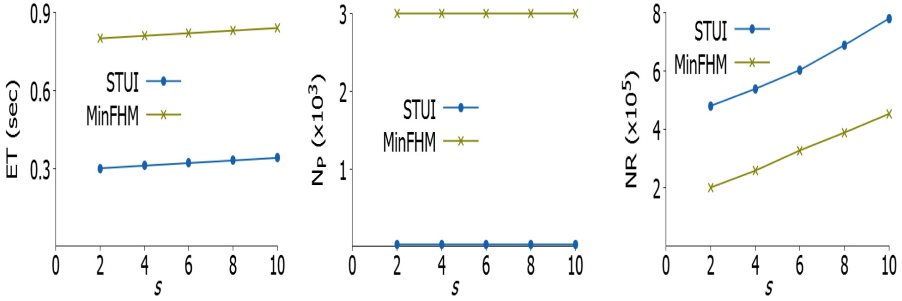
(a) Execution Time (b) Number of Patterns (c) Net Revenue

Fig. 6. Effect of variations in λ (Connect dataset)

high-revenue itemsets from its index. Notably, the results in Fig. 6 follow similar trends as those of Fig. 5; the actual values are lower in case of the results in Fig. 6 due to the smaller size of the Connect dataset w.r.t. the Retail dataset in terms of the number of items and the number of transactions.

5.3 Effect of Variations in *s*

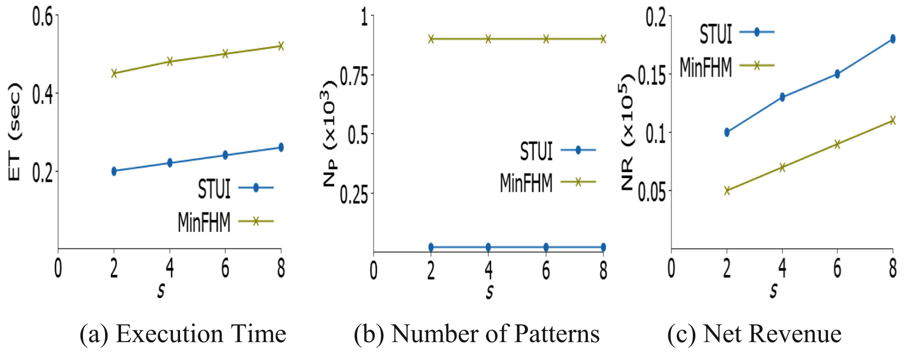
Figures 7 and 8 depict the results for the Retail dataset and the Connect dataset respectively when we vary the queried slot size *s*. The results in Figs. 7(a) and (b) indicate that STUI outperforms MinFHM in terms of ET and N_P due to the reasons explained earlier for the results of Figs. 5(a) and (b) respectively.



(a) Execution Time (b) Number of Patterns (c) Net Revenue

Fig. 7. Effect of variations in *s* (Retail dataset)

A detailed investigation of the experimental logs revealed that both ET and N_P increased albeit slightly for both of the schemes. This is because as the value of *s* increases, more slots need to be filled, thereby necessitating a slightly higher number of patterns to be examined. However, this increase in both ET and N_P is only slight for STUI because of its efficient indexing mechanism, which maintains the top-λ



(a) Execution Time (b) Number of Patterns (c) Net Revenue

Fig. 8. Effect of variations in *s* (Connect dataset)

high-utility itemsets. Thus, STUI only needs to examine the top-λ itemsets from its index level that corresponds to a given queried slot size; the slight increase arises from the traversal of more linked list entries at that level of the index in response to an increase in the queried slot sizes. On the other hand, the predominant cost for MinFHM arises from the candidate itemset generation, as discussed earlier. Recall that MinFHM has to generate all of the candidate itemsets of varied slot sizes first and then extract the itemsets of the queried slot size *s*. This explains the reason for the increase in both ET and N_P for MinFHM not being significant with increase in the queried slot size.

The results in Fig. 7(c) indicate that the net revenue NR increases for both the schemes with increase in *s*. This occurs because as the value of *s* increases, more slots need to be filled up. Hence, more items would be used to fill up an increased number of slots, thereby resulting in more revenue. STUI provides higher NR than that of MinFHM since STUI selects the top-λ high-revenue itemsets, while MinFHM randomly selects any λ itemsets, as explained for the results in Fig. 5(c). Observe that the results in Fig. 8 follow similar trends as those of Fig. 7; the difference in the actual values of the performance metrics arises due to the respective dataset sizes.

6 Conclusion

Utility mining has been emerging as an important area in data mining. Existing works on utility mining have primarily focused on the problem of finding high-utility itemsets from transactional databases. However, they implicitly assume that each item physically occupies only one slot. This is in contrast with many real-world scenarios, where the number of slots consumed by different items typically varies. Hence, in this paper, we have considered that a given item can occupy any fixed (integer) number of slots. In this regard, we have addressed the problem of efficiently determining the top-utility itemsets when a given number of slots is specified as input.

The key contributions of our work include an efficient framework to determine the top-utility itemsets for different queried slot sizes. Moreover, we have proposed the novel flexible and efficient STUI index for facilitating quick retrieval of the top-utility itemsets for a given number of slots. Furthermore, we have conducted an extensive performance evaluation using real datasets to demonstrate the overall effectiveness of the proposed indexing scheme in terms of execution time and utility (net revenue) as compared to a recent existing scheme. In the near future, we plan to perform additional experiments with more real datasets for enhancing the proposed framework before piloting and deploying the system in the real world.

References

1. Hansen, P., Heinsbroek, H.: Product selection and space allocation in supermarkets. Eur. J. Oper. Res. **3**, 474–484 (1979)
2. Yang, M.H., Chen, W.C.: A study on shelf space allocation and management. Int. J. Prod. Econ. **60–61**, 309–317 (1999)
3. Yang, M.H.: An efficient algorithm to allocate shelf space. Eur. J. Oper. Res. **131**, 107–118 (2001)
4. Chen, M.C., Lin, C.P.: A data mining approach to product assortment and shelf space allocation. Expert Syst. Appl. **32**, 976–986 (2007)
5. Chen, Y.L., Chen, J.M., Tung, C.W.: A data mining approach for retail knowledge discovery with consideration of the effect of shelf-space adjacency on sales. Decis. Support Syst. **42**, 1503–1520 (2006)
6. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: 20th International Conference, VLDB, pp. 487–499 (1994)
7. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. ACM Sigmod Record **29**, 1–12 (2000)
8. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Discovering frequent closed itemsets for association rules. In: Beeri, C., Buneman, P. (eds.) ICDT 1999. LNCS, vol. 1540, pp. 398–416. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-49257-7_25
9. World's largest retail store. https://www.thebalance.com/largest-retail-stores-2892923
10. US Retail Industry. https://www.thebalance.com/us-retail-industry-overview-2892699
11. Fournier-Viger, P., Wu, C.-W., Tseng, V.S.: Novel concise representations of high utility itemsets using generator patterns. In: Luo, X., Yu, J.X., Li, Z. (eds.) ADMA 2014. LNCS (LNAI), vol. 8933, pp. 30–43. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-14717-8_3
12. Fournier-Viger, P., Lin, J.C.-W., Wu, C.-W., Tseng, V.S., Faghihi, U.: Mining minimal high-utility itemsets. In: Hartmann, S., Ma, H. (eds.) DEXA 2016. LNCS, vol. 9827, pp. 88–101. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44403-1_6
13. Zida, S., Fournier-Viger, P., Lin, J.C.-W., Wu, C.-W., Tseng, V.S.: EFIM: a highly efficient algorithm for high-utility itemset mining. In: Sidorov, G., Galicia-Haro, S.N. (eds.) MICAI 2015. LNCS (LNAI), vol. 9413, pp. 530–546. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-27060-9_44
14. Fournier-Viger, P., Zida, S., Lin, J.C.-W., Wu, C.-W., Tseng, V.S.: EFIM-Closed: fast and memory efficient discovery of closed high-utility itemsets. In: Perner, P. (ed.) Machine Learning and Data Mining in Pattern Recognition. LNCS (LNAI), vol. 9729, pp. 199–213. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41920-6_15
15. Liu, M., Qu, J.: Mining high utility itemsets without candidate generation. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 55–64. ACM (2012)

16. Liu, Y., Liao, W.K., Choudhary, A.: A fast high utility itemsets mining algorithm. In: Proceedings of the 1st International Workshop on Utility-Based Data Mining, pp. 90–99 (2005)
17. Tseng, V.S., Wu, C.W., Shie, B.E., Yu, P.S.: UP-Growth: an efficient algorithm for high utility itemset mining. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 253–262. ACM (2010)
18. Tseng, V.S., Wu, C.W., Fournier-Viger, P., Philip, S.Y.: Efficient algorithms for mining the concise and lossless representation of high utility itemsets. IEEE Trans. Knowl. Data Eng. **27**, 726–739 (2015)
19. Chan, R., Yang, Q., Shen, Y.D.: Mining high utility itemsets. In: 3rd IEEE International Conference on Data Mining, ICDM, pp. 19–26 (2003)
20. SPMF (Open-source data mining library). http://www.philippe-fournier-viger.com/spmf/dataset
21. Fournier-Viger, P., Wu, C.-W., Zida, S., Tseng, V.S.: FHM: faster high-utility itemset mining using estimated utility co-occurrence pruning. In: Andreasen, T., Christiansen, H., Cubero, J.-C., Raś, Z.W. (eds.) ISMIS 2014. LNCS (LNAI), vol. 8502, pp. 83–92. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08326-1_9

Parallel Evolving Clustering Method for Big Data Analytics Using Apache Spark: Applications to Banking and Physics

Sk Kamaruddin[1,2], Vadlamani Ravi[1(✉)], and Pritman Mayank[1]

[1] Centre of Excellence in Analytics, Institute for Development
and Research in Banking Technology, Castle Hills Road No. 1, Masab Tank,
Hyderabad 500057, India
skkamaruddin@gmail.com, padmarav@gmail.com,
emailmayank2@gmail.com
[2] SCIS, University of Hyderabad, Hyderabad 500046, India

Abstract. A novel parallel implementation of the Evolving Clustering Method (ECM) is proposed in this paper. The original serial version of the ECM is the clustering method which computes online and with a single-pass. The parallel version (Parallel ECM or PECM) is implemented in the Apache Spark framework, which makes it work in real time. The parallelization of the algorithm aims to handle a dataset with large volume. Many of the extant clustering algorithms do not involve a parallel one-pass method. The proposed method addresses this shortcoming. Its effectiveness is demonstrated on a credit card fraud dataset (with size 297 MB), and a Higgs dataset was taken from Physics pertaining to particle detectors in the accelerator (with size 1.4 GB). The experimental setup included a cluster of 10 machines having 32 GB RAM each with Hadoop Distributed File System (HDFS) and Spark computational environment. A remarkable achievement of this research is a dramatic reduction in computational time compared to the serial version of the ECM. In future, the PECM shall be hybridized with other machine learning algorithms for solving large-scale regression and classification problems.

Keywords: ECM · Clustering · Big data analytics · Apache spark
Credit card fraud · Large Hedron Collider

1 Introduction

Clustering is the process of assembling objects having similarity in some aspect in the same group. There are several types of clustering methods viz. Centroid-based (partition based) clustering methods [1], Connectivity-based clustering (Hierarchical clustering) Methods, Agglomerative Approach [2], Divisive Approach [3], Density-based Clustering Method [4], Grid-based Clustering Method [5] and Model-based method [6]. The algorithms belonging to these methods are iterative in nature, i.e., making several passes over the data samples, thus rendering it unsuitable for big data analytics.

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 278–292, 2017.
https://doi.org/10.1007/978-3-319-72413-3_19

In centroid-based clustering methods, clusters are delineated by a central vector, called centroid of the cluster, which may not inevitably be an element of the dataset. The distance of the data points are calculated from the centroids of all the clusters, and the data point belongs to the cluster with minimum distance. These are iterative methods, which constitute several passes over the data.

The density-based clustering method searches for the dense regions in the data space. It differentiates different density regions and the data points within a given locality are considered to belong to the same cluster.

The grid-based clustering method divides the data space into a specific number of cells to construct a grid-like structure. Then, selecting dense regions from the cells in the grid structure results into the clusters.

All the clustering approaches mentioned above have shortcomings to handle large sized data with faster execution. Either they have the curse of dimensionality, or they are multi-pass algorithms, which render them inefficient for large datasets. Being online, the ECM can handle a stream of data while the clustering process goes on. Thus, the clustering process evolves with the incoming data points. Being one-pass and evolving in its approach, the ECM is suitable for large-sized dataset [7].

The afore-mentioned feature has provided a firm motivation for implementing a parallel version of ECM to handle massive dataset efficiently.

The remaining part of the article is ordered as follows: Apache Spark is introduced in Sect. 2. Section 3 describes related literature study. The problem definition is explored in Sect. 4. Section 5 describes the proposed methodology. The experimental setup is discussed in Sect. 6, and the details of the datasets are discussed in Sect. 7. The results and discussion are presented in the penultimate section. A conclusion and future directions are presented in Sect. 9.

2 Introduction to Apache Spark

Apache Spark is an open-source distributed in-memory computational framework for data analytics in the big data paradigm utilizing a cluster of commodity hardware, i.e., a group of affordable low-performance systems. Zaharia [8] developed Spark at UC Berkeley's AMPLab in 2009. It was an academic project in UC Berkley. Spark was meant to target interactive, iterative computations like machine learning. In the year 2013, the spark project was passed on to the Apache Software Foundation.

The Spark is comparably advantageous than other big data analytical technologies like Hadoop and Storm employing MapReduce framework. Spark is faster than MapReduce and offers low latency due to reduced disk input and output operation. Spark has the capability of in-memory computation, which makes the data processing faster than other MapReduce.

Unlike Hadoop, Spark maintains the intermediate results in memory rather than writing every intermediate output to disk. This operation hugely cuts down the execution time of the job, resulting in faster execution. When data crosses the threshold of the memory storage, it is spilled to the disk.

Spark uses data abstraction through the use of Resilient Distributed Dataset (RDD) for data processing [9]. Spark doesn't execute the tasks immediately but maintains a chain of operations as meta-data of the job called DAG (Directed Acyclic Graph) which are due to the transformation operation. The action on the DAG happens only when an 'action' operation is called on. This process is called as lazy evaluation. The lazy evaluation allows optimized execution of the queries on Big Data [10].

2.1 Spark Features

Apache Spark has other features, such as:

1. It supports a wide variety of operations compared to, Map and Reduce functions.
2. It presents a compact and stable Application Programming Interface (API) in the programming languages such as Scala, Java, and Python.
3. Spark is written in Scala Programming Language and runs in Java Virtual Machine (JVM).
4. An application can be developed employing any of the following programming languages: Scala, Java, Python, and R; in Spark.
5. It provides interactive programming interface called shell for Scala and Python.
6. It leverages the distributed cluster memory for doing computations for increased speed and data processing.
7. It runs on top of existing Hadoop cluster and access HDFS; it can also process data stored by HBase structure. It employs three different cluster managers for managing the resources of cluster viz. Yet Another Resource Negotiator (YARN) in Hadoop, Apache Mesos, and standalone mode.
8. Apache Spark can be integrated with various data sources like SQL, NoSQL, S3, HDFS, local file system, etc.
9. It is a good fit for iterative tasks like Machine Learning (ML) algorithms.
10. Apart from MapReduce computational framework, it supports SQL-like queries, streaming data, machine learning, and graph analysis.

2.2 Apache Spark Components and Architecture

Multiple applications run in Spark with independent resources and processes on a cluster. The main program or the driver program contains an object called SparkContext, which coordinates the applications.

For running an application on a cluster, the SparkContext connects to the cluster manager (i.e., either YARN, or mesos, or standalone). The cluster manager is responsible for resource allocation across applications. But, the standalone cluster can manage a single application only. Once Spark is connected to the cluster manager through the SparkContext, it acquires executors and resources for the executors on the worker nodes in the cluster. The executors are processes that perform computational work and storage of data for the application. SparkContext sends tasks to the executors to run (Fig. 1).

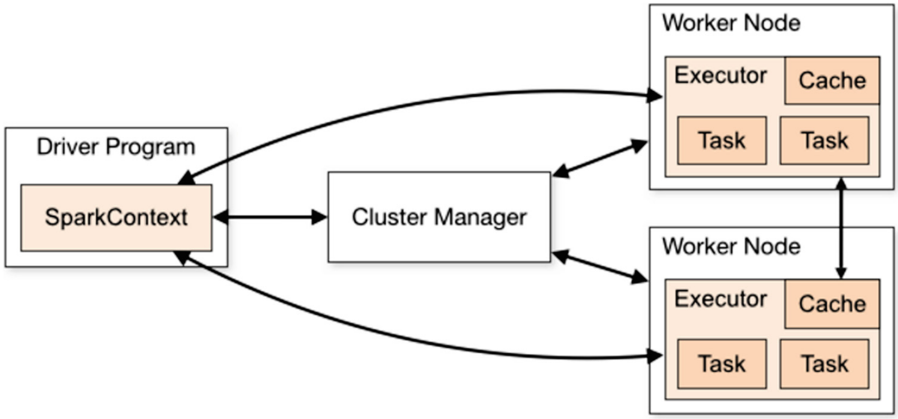


Fig. 1. Apache Spark components [11]

3 Literature Survey

There are several clustering approaches proposed by the research community. Based on the approach of clustering process they can be categorized into different categories pointed out in the introduction section. The literature survey was conducted with the articles about parallel implementation of different clustering algorithms.

The centroid-based clustering approach is the most explored one. The k-means algorithm is the frequently employed clustering algorithm, first proposed by Macqueen [1]. There are several endeavors from the research community to parallelize the k-means algorithm [12–16].

There are other centroid-based clustering approaches such as k-median, k-medoids, and fuzzy C-means. The parallel k-median algorithm has been proposed by Ene et al. [17]. The parallel version of k-medoid was proposed by Zhu et al. [18], and Jiang and Zhang [19]. Another clustering algorithm which has frequently been explored is fuzzy c-means [20, 21].

The density-based clustering method has an approach of separating dense regions from the sparsely populated regions to form the clusters. This method is adept in detecting clusters with randomness in their shape. One of the popular approaches for density-based clustering is DensityBased Spatial Clustering of Applications with Noise (DBSCAN) [4]. The parallel density-based clustering is proposed by Han et al. [22], and Chen et al. [23]. The parallel grid-based clustering method is also proposed by Chen et al. [23], and Gouineau et al. [24].

All the above-discussed algorithms are not suitable for a high dimensional large-sized dataset for online clustering process with one-pass execution. The ECM proposed by Song and Kasabov [7] is a one-pass online clustering method which is faster in comparison to the avant-garde parallel clustering approaches.

4 Problem Definition

The contemporary research community has proposed several clustering approaches. These approaches have either curse of the dimensionality or require multiple iterations to finalize the clusters. Thus, the contemporary clustering methods are not competent for analysis of the massive online data. Very few works involving parallel, distributed clustering utilizing Apache Spark have been reported in the past [25–27]. However, these are not online and incremental in nature, unlike ECM.

The ECM [7] is a clustering approach which updates the clustering process with each incoming data points in the data space. The ECM is effective and efficient for processing online data points.

Implementation of a fast one-pass execution process for clustering is the need of current research community to handle the clustering process for massive datasets. Hence, the current research work aims to develop a method for clustering of massive datasets.

5 Proposed Methodology

The proposed Parallel Evolving Clustering Method (PECM) is implemented in Apache Spark computational framework with distributed data storage in HDFS. Thus, the PECM algorithm executes in a parallel, distributed manner.

5.1 The ECM Algorithm

The algorithm of ECM is a distance-based clustering technique, which uses D_{thr}, a clustering parameter, which is the threshold value for similarity index. A sample point belonging to a cluster which is the farthest from the cluster center should be less than or equal to D_{thr}, the threshold value. D_{thr} influences the count of clusters that the algorithm yields. The ECM is presented below:

Step 1: Initialize the first cluster C_1 with the first data point from the input dataset and its position is considered as the cluster center Cc_1, for the cluster C_1 and the radius of the cluster C_1 is Ru_1, which is set to a value 0.
Step 2: If the analysis of entire samples from the dataset is completed then the clustering process comes to an end. Else, the present input instance, x_i, is considered and the normalized Euclidean distances *d(i, j)*, between this instance and the cluster centers Cc_j of all *n* already existing clusters,

$$d(i, j) = ||xi - Ccj||, \ where\ j = 1, 2, \ldots, n,$$

are evaluated. Here *d(a, b)* is the normalized Euclidean distance between the two *q*-dimensional vectors *a* and *b* and are defined as follows:

$$\|a - b\| = \left(\sum_{i=1}^{q} |a_i - b_i|^2\right)^{1/2} \Big/ q^{1/2} \tag{2}$$

where $a,\ b \in R^q$.

Step 3: The distance $d(i,\ m)$ between a sample x_i and a cluster center Cc_m where Cc_m is the center of a cluster C_m with radius Ru_m is defined as follows:

$$d(i,m) = \min_j d(i,j) = \min_j \left(\|x_i - Cc_j\|\right), \tag{3}$$

where, $j = 1, 2, \ldots, n$ and if $d(i,\ m) \le Ru_m$.

Then the current sample x_i is a member of the cluster C_m. In this occasion, no new cluster is formed. Also, no existing cluster is modified. The algorithm then goes back to Step 2.

Else,

Step 4: The extended distance $s(i,\ j)$ between sample x_i and cluster center Cc_j is evaluated by adding the distance value $d(i,\ j)$ and radius Ru_j of cluster C_j.

$$s(i,\ j) = d(i,\ j) + Ru_j,\ where\ j = 1, 2, \ldots, n \tag{4}$$

and then choosing the cluster C_a with the minimum value $s(i,\ a)$:

$$s(i,a) = d(i,a) + Ru_a = \min_j s(i,j), \quad \text{where } j = 1, 2, \ldots, n \tag{5}$$

Step 5: If $s(i,\ a) > 2D_{thr}$, the sample x_i cannot be a member of any existing clusters. A new cluster is formed as mentioned in Step 1. The algorithm then returns to Step 2.

Else,

Step 6: If $s(i,\ a) \le 2D_{thr}$, the cluster C_a is modified by relocating its center, Cc_a, and enlarging its radius value, Ru_a. The modified radius Ru_a^{new} is assigned the value equal to $s(i,\ a)/2$ and the new center Cc_a^{new} is positioned on the line joining input vector x_i and the former cluster center Cc_a so that the sample point x_i is at a distance of Ru_a^{new} from the newly formed center Cc_a^{new}. The algorithm then returns to Step 2.

5.2 The PECM Algorithm

The PECM algorithm is a parallel implementation of ECM algorithm. The PECM algorithm follows below:

Required: The dataset D has to be uploaded to HDFS for distributed storage. The dataset is normalized using min-max normalization.

Step 1: The data is divided into a specified number of partitions in the distributed storage to make the parallel execution efficient.

Step 2: The ECM algorithm is executed in all the partitions of the dataset, i.e., the number of instances of ECM is same as the number of partitions, running in parallel. The different partition of dataset produces clusters same as the number of classes for a given D_{thr} value.

Step 3: The sub-clusters are collected at the master node from each partition of data and merged in a parallel manner. The merging process is executed over the worker nodes in a parallel manner with partitioning of sub-clusters, i.e., the sub-clusters are divided into partitions, and each partition produces clusters same as the number of classes. The merging process is carried out with a parallel merging threshold. These clusters are collected and merged in the master node, which we call as the serial merging with a serial merging threshold thereby producing the required number of clusters.

The workflow of PECM is depicted in Fig. 2.

The map-reduce operation in Spark is executed with *transformation* and *action* operations. The PECM is executed in a parallel manner by mapping it to all the partitions of the dataset by a map operation e.g.

```
val clusterCentersRDD = partitionedData.mapPartitions(ECM)
```

where, the partitioned data is represented with *partitionedData* and *ECM* is the function passed to the *mapPartitions* function.

Then the reduce operation collects all the sub-clusters formed in each partition of the dataset. The reduce operation is carried out with *collect* action operation e.g.

```
val clusterCenters = clusterCentersRDD.collect()
```

Here, all the cluster centers are collected in *clusterCenters*.

According to the Step 1, initially, the dataset is divided into a number of partitions across the worker nodes, i.e., the data is stored in the HDFS spanning over the worker nodes. The partition number is chosen to be high enough using a trial and error method to reduce the latency. The count of ECM instances running in parallel is the same as the number of partitions of the dataset. In Step 2, depending on the $\mathrm{D_{thr}}$ value, a different number of sub-clusters are formed in each partition, which is the output of each instance of ECM.

Then, the sub-clusters are collected in the master node. The high number of partitions entails a high number of sub-clusters resulting from a high number of ECM instances running in parallel. Therefore, the situation warrants a parallel merging phase.

In the parallel merging phase, firstly, the collected sub-cluster centers are partitioned at the master node and are distributed over the worker nodes. Secondly, a merging process is executed on each partition of sub-cluster centers as follows. Two sub-clusters are merged if the distance between their centers is within a pre-specified merging threshold value. This process results in a new group of sub-cluster centers which are collected at the master node.

These new sub-cluster centers are then merged at the master node (which is called the serial merging phase) so that the final desired number of clusters is achieved.

In summary, the proposed PECM comprises (i) running as many instances of ECM in parallel as the number of partitions effected on the data (ii) a merging process involving a parallel phase at worker nodes and a serial phase at the master node in tandem. We observed that, without the invocation of merging process, the parallel and distributed version of ECM could not be developed. In other words, merging process

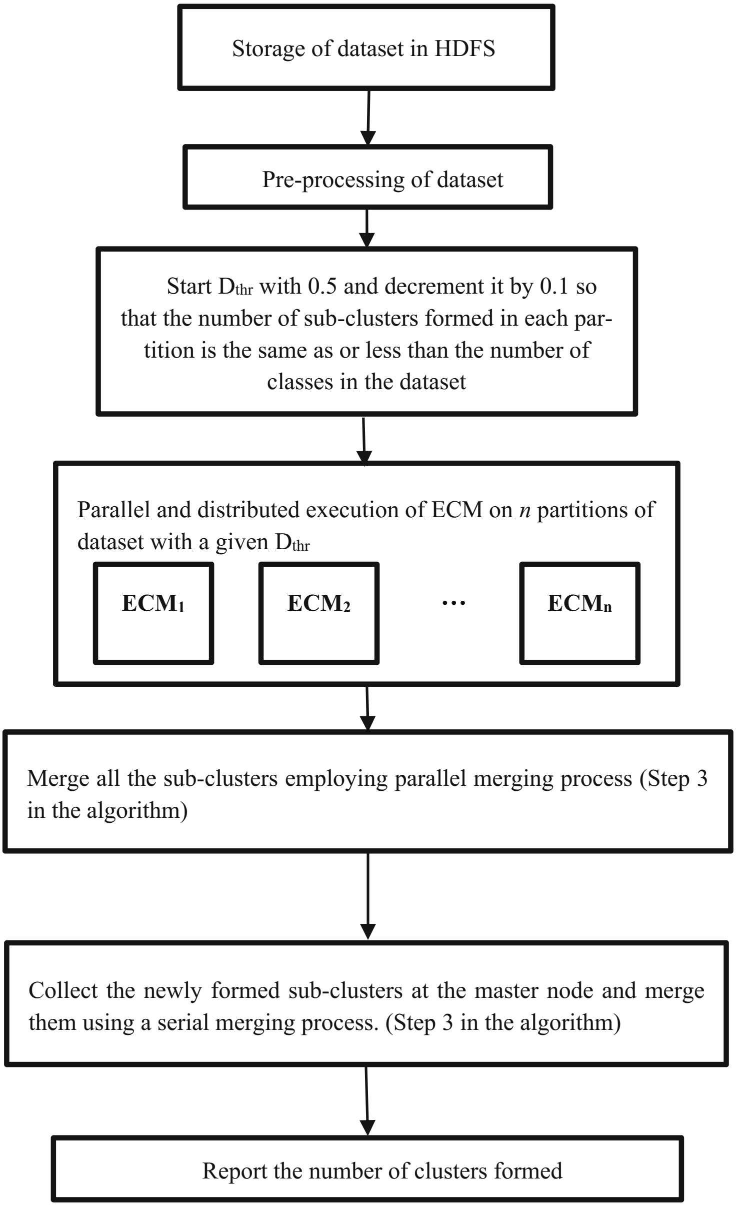

Storage of dataset in HDFS
Pre-processing of dataset
Start Dthr with 0.5 and decrement it by 0.1 so that the number of sub-clusters formed in each partition is the same as or less than the number of classes in the dataset
Parallel and distributed execution of ECM on n partitions of dataset with a given Dthr
ECM1
ECM2
…
ECMn
Merge all the sub-clusters employing parallel merging process (Step 3 in the algorithm)
Collect the newly formed sub-clusters at the master node and merge them using a serial merging process. (Step 3 in the algorithm)
Report the number of clusters formed

Fig. 2. Execution flowchart of PECM

helps PECM to reduce the huge number of sub-clusters formed of the data at various worker nodes. We believe that the proposed novel merging process is an important contribution to the literature of parallel incremental clustering techniques.

It may be noted that the total count of the clusters produced by the proposed PECM algorithm is the same as the number of class variables. If the class variable is not present in the dataset, then we can start with a random D_{thr} value and find the Dunn Index [28] of the clusters formed in each partition. Then, the sub-clusters from each partition with highest Dunn Index value obtained by varying the D_{thr} value will be collected at the master node for the serial merging process. Another way to resolve this problem is to visualize the data either in a multi-dimensional space using method of parallel coordinates [29] or a 3-dimensional space using principal component analysis plot [30]. Finally, while solving problems related to the business domain, one can take the help of domain expert to determine the number of clusters.

5.3 Computational Complexity of PECM

The ECM in step 2, calculates the distance of each sample from the existing clusters at that instant. In the worst case, each sample is compared with other samples for the distance calculation. This will result into $1 + 2 + \ldots + (n - 2) + (n - 1)$ number of distance calculation. Thus, distance calculation has the time complexity of $O(n^2)$, where n is the number of samples.

The PECM executes the ECM in a parallel manner in all the partitions thus, the time complexity of PECM is $O(n^2/p)$, where p is the total count of processors running the parallel execution of ECM [31].

Then the parallel merge process compares the elements of sub-clusters to be merged with a merging distance threshold. The worst case merging of two sub-clusters in a partition will take $O(n^2)$. The parallel execution in p processors will result in time complexity of $O(n^2/p)$.

Now, the serial merge will take the time complexity of $O(n^2)$. So, the total time complexity of PECM = $O(n^2/p) + O(n^2/p) + O(n^2) = O(n^2)$.

6 Experimental Setup

The experimental setup consists of standalone Spark cluster using the HDFS as the storage system and Apache Zeppelin 0.7.1 as an editor. The Spark cluster comprises a master node running the *driver program* and ten *worker nodes* including one worker node running on the *master node*. All the ten nodes have the same configuration, i.e., Intel® Core™ i7-6700 CPU @ 3.40 GHz with 8 logical cores with 32 GB RAM and Ubuntu 14.04 LTS Operating System (see Table 1). We allocated 28 GB of memory in nine worker nodes. In each of the worker node, four executors with 7 GB memory and two cores are configured. The worker in the master node is configured with three executors with 6 GB memory and two cores each. The driver process was allocated 10 GB of memory (see Table 2).

The PECM was executed in Spark 2.1.0 cluster with Hadoop 2.7.3 as distributed storage using Scala 2.11.8 programming language (see Table 2). The best execution time was achieved by tweaking the memory used by the executors in each worker node with the optimal number of data partitions.

Table 1. System description

CPU	Intel® Core™ i7-6700 CPU @ 3.40 GHz with 8 logical cores
Memory	32 GB
Operating system	Ubuntu 14.04 LTS

Table 2. Computational environment description

Configuration details	Node type	
	Master	Slaves
Driver memory	10 GB	–
Number of workers	3	4
Worker memory	6 GB	7 GB
Number of executors	3	4
Executor memory	6 GB	7 GB
Number of cores/executors	2	2
Total executor memory	18 GB	28 GB
Total memory utilized (out of 32 GB)	28 GB	28 GB
Computational framework	Apache Spark 2.1.0	
Distributed storage system	HDFS (Hadoop 2.7.3)	
Editor for code development	Apache Zeppelin 0.7.1	
Language used for coding	Scala 2.11.8	

7 Dataset Description

The description of the datasets follows here.

The credit card transaction count is growing day-by-day with leaps and bounds. There is a large volume of data generated through credit card transactions. The data is also generated at a high velocity. These conditions make the problem amenable to the application of big data analytics.

This dataset is a snapshot at a particular instant of time for processing, as there is non-availability of credit card dataset having a real-time inflow of transactions in the public domain.

The *ccFraud* dataset [32] contains ten million samples. We have made assumptions for the legitimate transactions as negative samples and fraudulent transactions as positive samples. The negative class has 9,403,986 samples. The positive class has 596,014 samples. The dataset contains nine features with a total size of 291.7 Mb. The different variables in the dataset are described in Table 3. The "*fraudRisk*" is a binary feature having 1 and 0 as two discrete values. Here, 1 represents a fraudulent transaction and 0 accounts for a non-fraudulent transaction. In the proposed model, seven features have been considered for clustering process of PECM. We discarded the "*custID*," since it contains unique values in all samples, which will disturb in the similarity calculation of the patterns. The class variable "*fraudRisk*" is also not included in the process of clustering to perform unsupervised learning.

Table 3. Details of *ccFraud* dataset

Feature name	Feature details
custID	Customer ID, an auto-incrementing integer value
gender	It takes two values either 1 or 2 for male and female, respectively
state	State number, an integer value
cardholder	It is the number of cards that belong to a customer, with a maximum value of 2
balance	The credit balance in the account
numTrans	Number of transactions made, an integer value
numIntlTrans	Number of international transactions made, an integer value
creditLine	Credit limit of a customer, an integer value
fraudRisk	It takes two values either 0 or 1 for genuine and fraudulent transactions, respectively

The other dataset, we analyzed is the Physics dataset generated from particle detectors in the accelerator, the HIGGS dataset [33]. The data is produced using Monte Carlo simulations. The dataset is nearly balanced with 53% positive samples in the dataset. The dataset contains 11000000 samples. In this, the first feature is the class variable with two values, i.e., 1 and 0 for representing signal and background respectively. The dataset contains 28 more features following the class variable out of which first 21 are low-level features, the next 7 are high-level features. The low-level features are kinematic attributes measured by the particle detectors in the accelerator. The low-level features are mapped to the high-level features. These high-level features are employed to define the class value.

The features of the HIGGS dataset are described in Table 4. We have utilized 7 high-level features for clustering purpose those are extracted from 21 low-level features for the PECM to cluster it into two clusters. The resulting dataset is 1.4 GB in size.

Table 4. Details of *HIGGS* dataset

Feature name	Feature details
lepton pT, lepton eta, lepton phi, missing energy magnitude, missing energy phi, jet 1 pt, jet 1 eta, jet 1 phi, jet 1 b-tag, jet 2 pt, jet 2 eta, jet 2 phi, jet 2 b-tag, jet 3 pt, jet 3 eta, jet 3 phi, jet 3 b-tag, jet 4 pt, jet 4 eta, jet 4 phi, jet 4 b-tag	21 low-level features. These are kinematic attributes
m_jj, m_jjj, m_lv, m_jlv, m_bb, m_wbb, m_wwbb	7 high-level features employed to distinguish between the two classes

8 Results and Discussion

The ccFraud dataset contains two class variables as legitimate transaction and fraudulent transaction. The PECM produced the clusters for each partition of the dataset present in the distributed storage system. These sub-clusters are then merged in a parallel as well as a serial manner to produce the final clusters. In the parallel merging process, the sub-clusters produced by PECM were merged using the Spark cluster with a parallel merging threshold. Then the result of parallel merging process was submitted to the serial merging process with its merging threshold. The selection of merging thresholds was automated to produce the clusters which are same as the number of class variables present in the dataset.

The ccFraud and Higgs dataset contains two class variables. Tables 5 and 6 presents the different combinations of parallel and serial merging thresholds those produced the desired clusters for ccFraud, and Higgs dataset, respectively. It may be noted that we can not present classification accuracy for the dataset because PECM is a clustering algorithm, which cannot output accuracy of prediction.

Table 5. Parallel and serial merging thresholds for ccFraud dataset

Merging threshold for parallel merging	Merging threshold for serial merging	No. of clusters formed
0.1	0.08	2
0.15	0.13	2
0.2	0.06	2
0.25	0.17	2
0.35	0.05	2

Table 6. Parallel and serial merging thresholds for Higgs dataset

Merging threshold for parallel merging	Merging threshold for serial merging	No. of clusters formed
0.15	0.05	2
0.2	0.18	2
0.25	0.05	2
0.3	0.08	2
0.4	0.06	2

The ccFraud dataset is having 94.04% of legitimate credit card transactions and only 5.96% of fraudulent transactions. Hence the dataset is highly unbalanced, yielding to the complexity involved in the clustering task.

The Higgs dataset is having 53% of positive samples and 47% of negative samples. Hence, the dataset is a balanced one leading to the simplicity of clustering process.

The PECM has a dramatic improvement over the serial ECM in terms of execution time. The ECM completed execution of ccFraud dataset in 74 s whereas PECM completed with 28 s. The ECM completed execution of Higgs dataset 77 s whereas PECM completed with 10 s (See Fig. 3). The speedup latency is calculated as follows:

$$S_{Latency} = \frac{L_{Serial}}{L_{Parallel}},$$

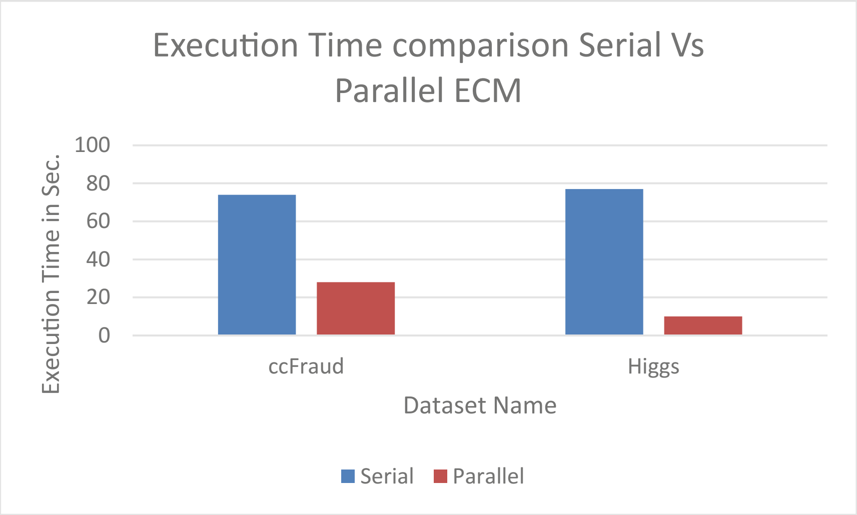


Fig. 3. Execution time comparison ECM Vs. PECM for ccFraud and Higgs dataset

where L_{Serial} is the latency due to serial execution, $L_{Parallel}$ is the latency due to parallel execution, and $S_{Latency}$ is the speedup.

Hence, $S_{Latency}$ for the *ccFraud* dataset is computed as follows:

$$S_{Latency} = \frac{L_{Serial}}{L_{Parallel}} = \frac{74\,\mathrm{s}}{28\,\mathrm{s}} = 2.6$$

Similarly, $S_{Latency}$ for *HIGGS* dataset is computed as follows:

$$S_{Latency} = \frac{L_{Serial}}{L_{Parallel}} = \frac{77\,\mathrm{s}}{10\,\mathrm{s}} = 7.7$$

Thus, 2.6x speedup for the ccFraud dataset and 7.7x speedup for Higgs dataset in comparison to serial execution is the remarkable achievement of the study.

Though the Higgs dataset is quite larger than ccFraud dataset, the execution of Higgs dataset is much faster than the ccFraud dataset. This discrepancy is attributed to the data distribution. The Higgs dataset is a balanced dataset whereas the ccFraud dataset is highly unbalanced one.

9 Conclusion and Future Works

The PECM is implemented with the Apache Spark with distributed storage of data points using HDFS. The PECM has achieved clustering of the large-sized dataset with a single go and thus is faster than any other clustering method. The PECM is implemented with Scala programming language. The ccFraud and Higgs datasets are analyzed.

An important contribution of the current work is the simple and innovative merging of sub-clusters involving both parallel and serial phases that is central to the distributed and parallel implementation of the ECM.

The performance of PECM is found to be 2.6x faster in the ccFraud dataset and 7.7x faster in case of the Higgs dataset.

Interestingly. the PECM can be combined with other machine learning techniques to solve various data mining tasks. For instance, PECM can be combined with (i) Probabilistic Neural Network (PNN) for the classification task; (ii) Generalized Regression Neural Network (GRNN) for regression task; (iii) Radial Basis Function Network (RBFN) for classification as well as regression tasks; (iv) Wavelet Neural Network (WNN) for classification and regression tasks. A fuzzy version of PECM is also proposed to be developed. Finally, PECM shall be scaled up to solve big data analytics problem with streaming data. These are the future directions in which we plan to work.

References

1. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium, vol. 1, pp. 281–297 (1967)
2. Murtagh, F.: A survey of recent advances in hierarchical clustering algorithms. Comput. J. **26**, 354–359 (1983)
3. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data, An Introduction to Cluster Analysis. Wiley, New York (1990)
4. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd, vol. 96, pp. 226–231 (1996)
5. Wang, W., Yang, J., Muntz, R.: STING: a statistical information grid approach to spatial data mining. VLDB **97**, 186–195 (1997)
6. Banfield, J.D., Raftery, A.E.: Model-based gaussian and non-gaussian clustering. Biometrics **49**, 803–821 (1993)
7. Song, Q., Kasabov, N.: ECM — a novel on-line, evolving clustering method and its applications. In: Foundations of Cognitive Science, pp. 631–682 (2001)
8. Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I.: Spark: cluster computing with working sets. HotCloud **10**, 95 (2010)
9. Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M.J., Shenker, S., Stoica, I.: Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing (2012)
10. Big Data Processing with Apache Spark – Part 1: Introduction. https://www.infoq.com/articles/apache-spark-introduction
11. Cluster Mode Overview - Spark 2.1.0 Documentation. https://spark.apache.org/docs/2.1.0/cluster-overview.html
12. Sun, Z., Fox, G., Gu, W., Li, Z.: A parallel clustering method combined information bottleneck theory and centroid-based clustering. J. Supercomput. **69**, 452–467 (2014)
13. Capó, M., Pérez, A., Lozano, J.A.: An efficient approximation to the K-means clustering for massive data. Knowl.-Based Syst. **117**, 56–69 (2017)
14. Liao, Q., Yang, F., Zhao, J.: An improved parallel K-means clustering algorithm with MapReduce. In: 2013 15th IEEE International Conference on Communication Technology, pp. 764–768. IEEE (2013)

15. Esteves, R.M., Hacker, T., Rong, C.: Competitive K-means: a new accurate and distributed K-means algorithm for large datasets. In: Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom, pp. 17–24. IEEE, Bristol (2013)
16. Lin, K., Li, X., Zhang, Z., Chen, J.: A K-means clustering with optimized initial center based on Hadoop platform. In: 2014 9th International Conference on Computer Science and Education, pp. 263–266. IEEE (2014)
17. Ene, A., Im, S., Moseley, B.: Fast Clustering using MapReduce Categories and Subject Descriptors. In: Kdd, pp. 681–689 (2011)
18. Zhu, Y.T., Wang, F.Z., Shan, X.H., Lv, X.Y.: K-medoids clustering based on MapReduce and optimal search of medoids. In: Proceedings of the 9th International Conference on Computer Science and Education, ICCCSE 2014, pp. 573–577 (2014)
19. Jiang, Y., Zhang, J.: Parallel K-medoids clustering algorithm based on Hadoop. In: 2014 IEEE 5th International Conference on Software Engineering and Service Science, pp. 649–652. IEEE, Beijing (2014)
20. Ludwig, S.A.: MapReduce-based fuzzy c-means clustering algorithm: implementation and scalability. Int. J. Mach. Learn. Cybern. **6**, 923–934 (2015)
21. Yu, Q., Ding, Z.: An improved Fuzzy C-means algorithm based on MapReduce. In: 2015 8th International Conference on Biomedical Engineering and Informatics (BMEI), pp. 634–638. IEEE (2015)
22. Han, D., Agrawal, A., Liao, W.-K., Choudhary, A.: A novel scalable DBSCAN algorithm with spark. In: 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 1393–1402. IEEE (2016)
23. Chen, C.C., Chen, T.Y., Huang, J.W., Chen, M.S.: Reducing communication and merging overheads for distributed clustering algorithms on the cloud. In: Proceedings of the 2015 International Conference on Cloud Computing and Big Data, CCBD 2015, pp. 41–48 (2016)
24. Gouineau, F., Landry, T., Triplet, T.: PatchWork, a scalable density-grid clustering algorithm. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing - SAC 2016, pp. 824–831. ACM Press, Pisa (2016)
25. Tsapanos, N., Tefas, A., Nikolaidis, N., Pitas, I.: Distributed, MapReduce-based nearest neighbor and E-ball kernel k-means. In: 2015 IEEE Symposium Series on Computational Intelligence, pp. 509–515. IEEE, Cape Town (2015)
26. Ketu, S., Agarwal, S.: Performance enhancement of distributed K-means clustering for big Data analytics through in-memory computation. In: 2015 Eighth International Conference on Contemporary Computing (IC3), pp. 318–324. IEEE, Noida (2015)
27. Tsapanos, N., Tefas, A., Nikolaidis, N., Pitas, I.: Efficient MapReduce kernel k-means for Big Data clustering. In: Proceedings of the 9th Hellenic Conference on Artificial Intelligence - SETN 2016, pp. 1–5. ACM Press, Thessaloniki (2016)
28. Dunn, J.C.: A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. J. Cybern. **3**, 32–57 (1973)
29. Inselberg, A.: Parallel coordinates. In: Liu, L., Özsu, M.T. (eds.) Encyclopedia of Database Systems, pp. 2018–2024. Springer, Boston (2009). https://doi.org/10.1007/978-0-387-39940-9_262
30. Pearson, K.: On lines and planes of closest fit to systems of points in space. Philos. Mag. **2**, 559–572 (1901)
31. Roosta, S.H.: Parallel Processing and Parallel Algorithms: Theory and Computation. Springer, New York (2000). https://doi.org/10.1007/978-1-4612-1220-1
32. ccFraud dataset. https://packages.revolutionanalytics.com/datasets/
33. Baldi, P., Sadowski, P., Whiteson, D.: Searching for exotic particles in high-energy physics with deep learning. Nat. Commun. **5**, 1–9 (2014)

An Introduction to Adversarial Machine Learning

Atul Kumar[1](✉), Sameep Mehta[2], and Deepak Vijaykeerthy[1]

[1] IBM Research, G2 Block, 8th Fl., Manyata Tech Park,
Ngawara, Bangalore 560045, India
{kumar.atul,dvijayke}@in.ibm.com

[2] IBM Research, ISID Campus, Institutional Area,
Vasant Kunj, New Delhi 110070, India
sameepmehta@in.ibm.com

Abstract. Machine learning based system are increasingly being used for sensitive tasks such as security surveillance, guiding autonomous vehicle, taking investment decisions, detecting and blocking network intrusion and malware etc. However, recent research has shown that machine learning models are venerable to attacks by adversaries at all phases of machine learning (e.g., training data collection, training, operation). All model classes of machine learning systems can be misled by providing carefully crafted inputs making them wrongly classify inputs. Maliciously created input samples can affect the learning process of a ML system by either slowing the learning process, or affecting the performance of the learned model or causing the system make error only in attacker's planned scenario. Because of these developments, understanding security of machine learning algorithms and systems is emerging as an important research area among computer security and machine learning researchers and practitioners. We present a survey of this emerging area named Adversarial machine learning.

Keywords: Adversarial learning · Computer security · Intrusion detection

1 Introduction

Over last few years, machine leaning has become a prominent technological tool in several application areas such as computer vision, speech recognition, natural language understanding, recommender systems, information retrieval, computer gaming, medical diagnosis, market analysis etc. In many areas, it is no longer a promising but immature technology as machine learning based systems have reached close to human level performance. Most of machine learning techniques build models using example data (training data). These models along with algorithms can be used to make predictions on data not seen before.

Learning and building models using training data provides hackers opportunities to attack machine learning algorithms by playing with the features and decision boundaries of the model. An adversary can craft malicious inputs to attack the performance or

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 293–299, 2017.
https://doi.org/10.1007/978-3-319-72413-3_20

efficiency of a machine learning algorithm. They can dupe an already trained system by creating input data that exploits the system into making glaring errors.

For example, researchers have demonstrated [1], ways to fool an image classification system by making tiny changes to the input images. Figure 1 shows several sets of examples with three images in each set. In each set, on the left is an image that the system correctly classifies. Image in center is a noise which when added to the left image creates an image shown on the right which still looks like the image on left to a human observer. But the system now classifies the image on right as an ostrich for every example. These techniques can be used by hackers to evade the system in making it accept malicious content as a genuine one. With machine learning becoming an important tool in strategically important applications such as security surveillance and background check for visa decisions etc., it is important to understand these attacks and make machine learning algorithms more robust against these attacks.

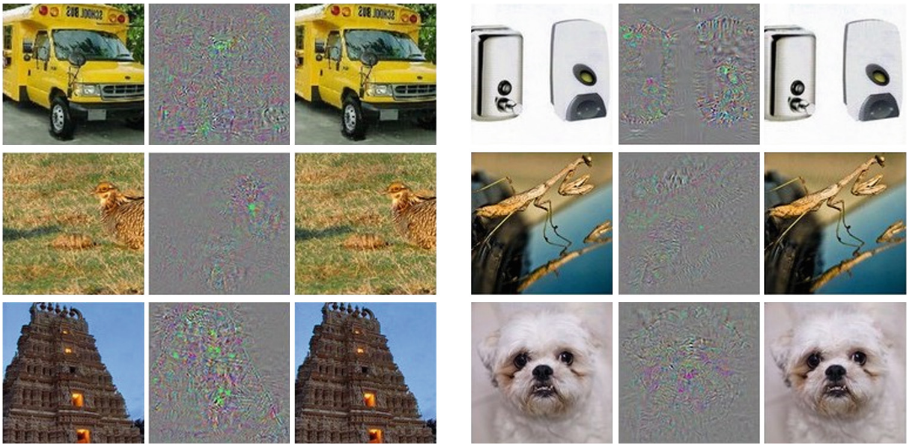


Fig. 1. Creating adversarial examples using noise (Image credit: Szegedy et al. [1])

If a hacker does not already know the algorithm, he first tries to learn the algorithm and its underlying model (e.g., logistic regression, neural network, decision trees etc.). Sometime, the hacker may only be interested in learning the model so that he can build his own ‘copy’ of the system using the learned model. This may be useful if the application is offered as a service via APIs and users are charged per use of these APIs. A hacker can create a sequence of inputs and then by observing outputs of the system corresponding these inputs, he can build a local model that may be very close to the model used by the original system. Depending on the pricing and the license terms of the API usage, a hacker may be able to ‘steal’ the model using very small amount of money. Tramer et al. demonstrated at USENIX Security Symposium 2016 [2] that models can be extracted from popular online machine learning services such as BigML and Amazon Machine Learning with a relatively small number of API calls.

Another category of attacks on machine learning systems is to provide adversarial input during the training phase and compromise the learning by affecting its efficiency or introducing some bias. Many systems allow users to provide training data samples for online training of the system. Collecting training data from people spared across geographies is immensely valuable in many applications to have good data distribution. But opening the system to public for providing input data also opens a system to malicious input created by hackers to 'poison' the system. Microsoft's twitter chatbot Tay started tweeting racist and sexist tweets in less than 24 h after it was opened to public for learning [3].

Attacks on Machine Learning based systems can be categorized in three broad categories. First set of attacks called exploratory attacks, try to 'steal' the algorithms and their models or some insight into the training data by providing carefully crafted inputs and then observing the output to build local copies of the models. Second set of attacks called evasion attacks, consists of techniques focusing on evading a system by making it classify an input incorrectly. And the third type of attacks called poisoning attacks, try to change the model of the system by providing malicious training examples aiming to alter the model of the machine learning algorithm.

2 Exploratory Attacks

Exploratory attacks do not attempt to influence training; instead they try to discover information from the learner that includes discovering which machine learning algorithm is being used by the system, state of the underlying model and training data.

2.1 Model Extraction Using Online APIs

Machine learning as a service for applications such as predictive analytics are deployed with publicly accessible query interfaces (APIs). These models are deemed confidential due to their sensitive training data, commercial value, or other reasons such as use in security applications. Access is provided on a pay-per-query basis. In such situations, an adversary has black-box access but no prior knowledge of the machine learning model's parameters or training data.

Tramer et al. [2] presented simple attacks to extract target machine learning models for popular model classes such as logistic regression, neural networks, and decision trees. Model extraction attacks were demonstrated on popular online ML-as-a-service providers such as BigML and Amazon Machine Learning. Their attacks were complete black box and the adversary does not even need to know the model type or any distribution information about training data. They could build local models that are functionally very close to the target. In some experiments, their attacks extracted the exact parameters of the target (e.g., the coefficients of a linear classifier or the paths of a decision tree). In situations where the model type, parameters or features of the target were not known, they used an additional preliminary attack step to reverse-engineer these model characteristics. Machine learning prediction APIs of major online services such as Google, Amazon, Microsoft, and BigML all return precision confidence values along with class labels. Moreover, they work with partial queries lacking one or more features. These features can be exploited for model extraction attacks.

3 Evasion Attacks

Evasion attacks are the most prevalent type of attack on a machine learning system. Malicious inputs are carefully crafted to evade detection which essentially means that input is modified to make the machine learning algorithm classify it as a safe one instead of malicious.

3.1 Adversarial Examples

Szegedy et al. [1] found that deep neural networks (DNN) learn input-output mappings that are fairly discontinuous. One can cause a DNN to wrongly classify an image by applying a specifically crafted modification (found by maximizing the network's prediction error) that is difficult to distinguish by a human viewer. The same change to the image can cause a different network, trained on a different subset of the dataset, to incorrectly classify the same image. This property of deep neural network can be exploited to create any number of adversarial inputs from the normal inputs.

Practical Black-Box Attacks method proposed by Papernot et al. [4] misclassified 84.24% of the crafted adversarial examples on MetaMind (an online deep learning API) DNN. They also used logistic regression substitutes to craft adversarial examples for Amazon and Google ML APIs and found misclassification rate of 96.19% and 88.94% respectively.

Papernot et al. [5] show that adversarial attacks are also effective when targeting neural network policies in reinforcement learning. Adversaries capable of introducing small perturbations to the raw input can significantly degrade test-time performance. The strategy is to train a local substitute DNN using a synthesized data set. Input data is synthesized but the label assigned is what the target DNN assigns to it and observed by the adversary. Adversarial examples are generated by using the substitute parameters known to adversary. These are misclassified by both target DNN and the substitute DNN created locally because they both have the same decision boundaries. To create a small perturbation so that the changed image looks similar to the original one, an algorithm named fast gradient sign method [6]. The cost gradient is computed for pixels and the target pixels (areas) for perturbation is identified. Another algorithm by Papernot et al. [7] can cause a misclassification for samples from any legitimate source class to any chosen target class. That is, any image can be changed slightly such that it is classified to a desired class (say ostrich) by the DNN. Therefore, a school bus image can be changed in such a way that to humans, it still looks like a bus but the DNN recognizes it as an ostrich (for that matter any class chosen by the adversary). Input components are added to a perturbation in order of decreasing adversarial saliency value until the resulting adversarial sample is misclassified by the model.

3.2 Generative Adversarial Networks (GANs)

Goodfellow et al. [8] introduced Generative adversarial networks. They are implemented by simultaneously training two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This can be viewed as a competition between a team of counterfeiters and a team of police. If generative model is assumed to be producing fake currency such that it can pass without detection, then the discriminative model is trying to detect the counterfeit currency. Competition leads both teams to improve their methods until the counterfeits cannot be distinguished from the genuine currency.

3.3 Evasion Attacks on Text Classification Systems

Perturbation techniques for image or audio based system cannot directly work on text based systems. That is because an important requirement of the perturbation used is to change the image or audio such that it still looks good to a human observer/listener. Whereas in a text, changing words by adding/deleting characters or changing sentences by adding/deleting words may make the sentence/word meaningless or change its meaning significantly and therefore cannot remain unnoticed by a human reader. Therefore, a perturbation technique must change the text such that it still looks good/suspicious to a human observer but machine learning system fails to classify it correctly after perturbation. For example, a spam email carrying an advertisement should still carry the advertisement message but fool the spam filtering system in classifying it as a regular email.

Creating adversarial inputs for text classification systems seems to be a harder problem than doing the same for the image or audio classification. Some recent work has shown that it is possible to systematically create such adversarial inputs. Liang et al. [11] discuss the problem of creating perturbation. They propose three techniques named insertion, modification, and removal, to generate adversarial samples for given text. They compute cost gradients (originally proposed in [6] for images and proven to be effective in [7, 12]) to decide what and where should be inserted, what and how to modify and what should be removed from a text sample. However, using the fast gradient sign method (FGSM) of [6] directly makes the text unreadable. Using cost gradient, they identify the text items that possess significant contribution to the classification. Then instead of changing the characters arbitrarily, they use one or more of insertion, modification and removal to craft an adversarial sample for a given text.

4 Poisoning Attacks

In poisoning attacks, attackers try to influence training data to influence the learning outcome. The purpose of poisoning attacks may vary from affecting the performance of learning algorithm to deliberately introducing specific biases in the model. In many applications, training is not a one-time job and model is often retrained to accommodate for the change in data distribution. In some situation, data collection is crowd-sourced and many users provide data sample that are used to continuously train the model. Some domains such as network intrusion detection, spam filtering, malware detection etc. are highly suspect of poisoning attacks but any machine learning system can be a victim of poisoning attacks.

4.1 Defensive Distillation

Papernot et al. [9] introduced a defensive mechanism called defensive distillation that reduces the effectiveness of adversarial samples on deep neural networks (DNNs). Distillation is a training procedure that was designed to train a DNN using knowledge transferred from a different DNN [10]. The motivation behind the knowledge transfer is to reduce the computational complexity of DNN architectures by transferring knowledge from larger architectures to smaller ones. This facilitates the deployment of deep learning in resource constrained devices that cannot rely on powerful GPUs to perform computations. A new variant of distillation is proposed for defense training. Instead of transferring knowledge between different architectures, knowledge extracted from a DNN is used to improve its own resilience to adversarial samples. An analytical investigation is presented for the generalizability and robustness properties granted by defensive distillation when training DNNs. Two DNNs were placed in adversarial settings to empirically study the effectiveness of defensive distillation. They show that defensive distillation can reduce effectiveness of sample creation from 95% to less than 0.5% on the DNNs used in their study. This can be explained by the fact that distillation reduces by a factor of 1030 the gradients used in adversarial sample creation. Distillation also increases by 800% the average minimum number of features required to be modified for creating adversarial samples on one of the DNNs used in their experiments.

References

1. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing Properties of Neural Networks. https://arxiv.org/pdf/1312.6199v4.pdf
2. Tramer, F., Zhang, F., Juels, A., Reiter, M.K., Ristenpart, T.: Stealing machine learning models via prediction APIs. In: USENIX Security Symposium (2016)
3. Reuters: Microsoft's AI Twitter bot goes dark after racist, sexist tweets, 24 March 2016. http://www.reuters.com/article/us-microsoft-twitter-bot-idUSKCN0WQ2LA
4. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: ACM Asia Conference on Computer and Communications Security (ASIACCS), April 2017
5. Papernot, N., McDaniel, P., Goodfellow, I.: Transferability in Machine Learning: From Phenomena to Black-Box Attacks using Adversarial Samples. https://arxiv.org/pdf/1605.07277.pdf
6. Goodfellow, I., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (ICLR) (2015)
7. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: IEEE European Symposium on Security and Privacy (Euro S&P) (2016)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Networks. https://arxiv.org/pdf/1406.2661.pdf
9. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: IEEE Symposium on Security and Privacy (SP) (2016)

10. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: Deep Learning and Representation Learning Workshop at NIPS (2014). https://arxiv.org/pdf/1503.02531.pdf
11. Liang, B., Li, H., Su, M., Bian, M., Li, X., Shi, W.: Deep Text Classification Can be Fooled. arxiv: https://arxiv.org/abs/1704.08006
12. Moosavi-Dezfooli, S-M., Fawzi, A., Frossard, P.: DeepFool: a simple and accurate method to fool deep neural networks. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

Investigating the Role of Twitter in E-Governance by Extracting Information on Citizen Complaints and Grievances Reports

Swati Agarwal[1] and Ashish Sureka[2](✉)

[1] Birla Institute of Technology and Science, Pilani, Goa, India
swatia@goa.bits-pilani.ac.in
[2] Ashoka University, Sonepat, Haryana, India
ashish.sureka@ashoka.edu.in

Abstract. Open Source Social Media Intelligence (OSSMInt) is a field that focuses on extracting useful information and actionable insights from publicly available and overt sources of data on social media platforms. There are several applications that can be built by applying OSSMInt techniques on this human-sensor data. In this paper, we present some of the use-cases of OSSMInt that are useful for the public sector agencies for e-governance. E-governance on social media include the identification of complaints and grievances reported online by the public citizens for the government authorities and facilitate public agencies to response those complaints, provide better services and improve their connections with public citizens. We present the basic Natural Language Processing and Machine Learning based framework, tools and techniques within the context of OSSMInt and E-governance. The focus of this paper is on mining user-generated content on Twitter (the most popular social media and microblogging website) to identify public citizens complaints and grievances. In particular, we focus on two important applications: (1) complaints which are reported to spread awareness among other citizens and to bring government's attention to the issues reported in the complaint, and (2) complaints which seek for immediate action and response from the concerned authorities. In addition to the basic introduction and motivation, we will discuss the unique challenges to these applications, open research problems, important literature, proposed approach, experimental results, and future directions.

Keywords: Bad roads complaints · Complaints and grievances
Government applications · Information visualization
Lexical knowledgebase · Mining user generated content
Natural Language Processing · Social media analytics
Text classification · Twitter

1 Introduction

Recently, there has been a noticeable adoption of social media and upward trend in its usage by government agencies for not just disseminating information but

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 300–310, 2017.
https://doi.org/10.1007/978-3-319-72413-3_21

also acquiring information such as complaints and grievances from citizens (a phenomenon referred to as *citizensourcing*) [8,12]. Specifically, social media websites like Twitter and Facebook are gaining popularity as social-media based citizen grievance management system or platforms on which people can lodge complaints. Evidence shows that Twitter is the most widely used micro-blogging platform on Internet. Due to the wide reachability and connectivity among its users, Twitter is being used by the National Government to reach out the public. For example, in India, ministry of the railway (@railminindia), state police (@delhipolice and @mumbaipolice), ministry of road and transport (@morthindia), traffic police (@dtptraffic) and income tax department (@incometaxindia) have some of the most active Twitter accounts. Unlike other countries, one of the primary objectives of Indian Government's Twitter accounts is to not only reach out to the public but to also address their complaints and grievances[1,2]. Based on our analysis of several Indian Government Twitter accounts, we find that an active Government Twitter handle receives an approximate of 5 tweets per minute. Based on our analysis of several Indian governments' Twitter handle data, we found that 50% of the tweets posted in an hour are complaints and grievances reported from various regions of India. The government bodies on Twitter forward these complaints and redirect authors to the concerned department for resolving their complaints efficiently.

We conduct a literature survey in the area of complaints and grievances identification on social media and divide the existing literature into two lines of research: (1) **usage of Twitter microblogging website to report and complaints and grievances** Heverin et al. [11] examine the use of Twitter by city police departments in large U.S. cities (cities with populations greater than 300,000) that have active Twitter accounts. Anderson et al. [5] present a study on Twitter adoption across American municipal police departments serving populations over $100,000$. Meijer et al. [15] present an empirical analysis of Twitter usage by the Dutch police and conduct an analysis of 982 Twitter handle. Edwards et al. [7] present a study on webcare, i.e. the act of engaging in online communication with citizens. Vanessa et al. [8] present a study for analyzing the behavioral similarities and differences of 3-1-1 phone service (formal) and Twitter (informal) channels for reporting issues in the community. (2) **mining public complaints and communication on Twitter for building prediciton models for situation awareness.** Kumar et al. [13] present an application of Twitter to atomically determine road hazards by building language models based on Twitter users' online communication. Gu et al. [10] propose a methodology to mine tweet texts to extract incident information on both highways and arterial roads. Fu et al. [9] describe an approach to extract and analyze real-time traffic related Twitter data for incident management purpose. Eleonora [6] present a real-time monitoring system for traffic event detection from Twitter stream analysis and conduct a case-study for the Italian road network. Mai et al. [14] demonstrate the use of data from public social interactions on Twitter as a

[1] http://bit.ly/2fR7R1s.

[2] http://bit.ly/2xLyEpN.

potential complement to traffic incident data. Schulz et al. [17] present a solution for a real-time identification of small-scale incidents using microblogs, thereby allowing to increase the situational awareness by harvesting additional information about incidents. Napong et al. [18] present a study on social-based traffic information extraction and classification consisting of mining Twitter for traffic congestion, incidents, and weather. Furthermore, in order to address the complaints and grievances of citizens, the Indian government also initiated several policies and organizations. The aim of these organizations (TwitterSeva, MociSeva, Cybercell, DOTSeva) is to mine online complaints specific to the concerned department and address them in a timely and efficient manner.

The research work presented in this paper is motivated by the need to develop a solution to automatically resolve the challenges of manual inspection. Twitter allows users to post a maximum of 140 characters and therefore involves usage of slang and abbreviations. Due to the presence of free-form text tweets do not have a defined structure or language format and hence are high likely to have grammar and spelling errors. Due to the presence of multilingual texts and scripts in tweets, it is challenging to identify the linguistic features for building NLP (Natural Language Processing) based applications. Text classification or categorization and information extraction from tweets is thus a technically challenging problem [2–4]. Further, filtering these complaint reports from non-complaint tweets is technically challenging due to the wide range of complaints. Our research aim is to build a text analysis based model to address the NLP challenges in microposts (very short text such as tweets). In particular, the research aim of the work presented in this paper is to investigate text classification based techniques for automatically identifying complaints tweets and assigning them to predefined labels based on the topic of the content. Our research aim is also to investigate information extraction and visualization to extract useful information and insights from the complaint reports. Furthermore, our aim is to create an annotated dataset and make it publicly available to the research community.

2 Data Collection and Pre-processing

We formulate the problem of automatic identification of citizens' complaints (reported to official Twitter handle of a public agency) as a one-class classification problem. We propose a text classification based approach consisting of various components performing several tasks: tweets extraction from public agencies' account, enrichment and enhancement of raw microposts (tweets), learning the features of non-complaint and complaint report tweets, developing a baseline classification approach, use of ensemble techniques to improve the baseline method, empirical analysis and performance evaluation. Based on our inspection on complaints and grievances reports on Twitter, we divide them into two case studies (as discussed in the literature survey. See literature survey in Sect. 1). Due to the page limit, we present the data collection and statistics in form of a table. Tables 1 and 2 demonstrate the statistics of dataset for Case Study 1 (4 weeks duration-11 April 2016 to 8 May 2016) and Case Study 2 (8 weeks-from

Table 1. Dataset for case study 1

Account	Original	Sampled
@RailMinIndia	36182	1500
@dtpTraffic	1524	1000
@DelhiPolice	1720	1000
@IncomeTaxIndia	383	200

Table 2. Dataset for case study 2

@nitin_gadkari, @MORTHIndia		
Collected	Original	Replied
81304	17511	11092
Retweets	Sampled English tweets	Users
52701	3302	2604

July 18, 2016 to September 13, 2016) respectively. Tables 1 and 2 shows the number of tweets collected, and sampled (after pre-processing and filtering) for our experiments. The statistics shown in the tables also shows the name of the public agencies' accounts for which we collected the data.

We preprocess the sampled datasets (for both case study 1 and 2) and address the challenge of noisy content in the tweets. We use the micropost enrichment algorithm proposed in our previous study Mittal et al. (focused on the problem of complaints and grievances identification) [16]. The proposed algorithm performs a syntactic enhancement of the tweets and consists of five phases primarily named as hashtag expansion (splitting the joint hashtags), @username mention expansion (replacing the @mentioned usernames with their actual profile names), spell error correction (using the application of Bing Search Engine to correct the spelling error in tweets), acronym and slang treatment (correcting the domain specific and generalized slang and abbreviations in tweets) and sentence segmentation (removing filler terms, consecutive special characters, correcting spaces and conjunctions).

3 Identifying Non-complaints Reports

The official public service agencies account on Twitter are open and anyone can mention them in their tweets. We observe that not all tweets posted on these accounts are complaint reports and rather are either off-topic or discussions not relevant to the complaint department. In order to classify a complaint report related to a government service (case studies 1 and 2), we identify various features that are a strong indicator of a tweet to be a complaint report. However, we also observe that there are several discriminatory features that are a strong indicator of a tweet certainly not to be complaint report. Based on our manual inspection on official Twitter handle and non-complaint tweets identification model proposed in our previous study [16], we divide such tweets into 4 categories (AISP): Appreciation posts, Information Sharing and Promotional tweets.

Appreciation tweets are the posts made by citizens for praising the government for their work or resolving their previous complaints. Information sharing tweets are the tweets posted by users to share daily news about the events or policies initiated by the government. Further, due to the presence of several official accounts on Twitter, we categorize tweets posted by a different official

Table 3. Examples of non-complaint tweets-classified into 4 categories: appreciation (APP), information sharing (IS) and promotional (PRL) tweets.

AISP	Tweet
APP	Big **big move** @mansukhmandviya ji. **Thank you** for personally monitoring this. **Thanks** to @MORTHIndia and @nitin_gadkari ji also for this
IS	Commuters, get ready for more sun-kissed rides on the waters https://goo.gl/SlrCWE @PiyushGoyal @nitin_gadkari
PRL	Income declaration scheme: Government assures complete confidentiality: http://goo.gl/jVU5qK @FinMinIndia **@IncomeTaxIndia**

account of same public agencies categorized as promotional tweets. Table 3 shows examples of appreciation, information sharing, and promotional tweets posted on official accounts of @nitin_gadkari and @MORTHIndia. We use the AISP tweet classifier method proposed in our previous study [16] and classify AISP tweets and filter the unknown posts that may or may not be a complaint report about killer roads. We use these unknown tweets for further feature extraction and classification and complaints reports classification.

4 Features Extraction

In this Section, we identify various linguistic and contextual features that can be used to classify a complaint tweet. Due to the page limit, we present all features in form of a table. Tables 4 and 5 shows the list of all features extracted for

Table 4. List of features extracted in case study 1. CDK = Closed Domain Keywords

Feature	Summary	Technique	Presentation
N-Grams	Character n-grams frequently occurring in complaint posts	WordNet	Grouped Triplets of n-grams similar to each other
CDK	Keywords specific to different departments of public agencies	Manual Inspection	Each word is a column in feature vector space
Events	Activities, events and substances reported in the complaints	IBM Watson Bluemix Concept and Relationship (ICR)	Each substance is a column in feature vector space
Location	Location of the incident or reported complaint	ICR, Google Geocoding API	Location is an attribute in the feature space
Media	Presence of multimedia files such as video or images	Twitter API	A boolean vector in the feature space denoting the presence or absence of media in a tweet

Table 5. List of features extracted in case study 2.

Feature	Summary	Technique	Presentation
Problem	The issue reported in the complaint for prompt addressal by concerned department	CoreNLP, POS Tagging, ConceptNet	Problem or issue reported in a tweet is a unique vector in feature space
Location	The region (or city) from where the complaint is reported	Indico API	Location is an attribute in our column-divided into city, state, town, region
Landmark	The exact pinpoint location (or landmark) of the problem	OpenStreetMap API	Landmark is a unique vector in our feature space

case study 1 and case study 2 respectively. Since this tutorial is compiled from our previous papers, in this paper, we only provide the summary of extracted features. We recommend our readers to read the full version of our previous studies Mittal et al. [16] and Agarwal et al. [1].

5 Classification

5.1 Case Study 1

In next step of the processing pipeline of our proposed solution method, we use an ensemble learning based Support Vector Machine (SVM) classifier. We divide our data (tweets categorized as unknown in AISP classifier) into training (25%) and testing dataset (75%). We use the feature vector model created in previous phase and learn our one-class classification model-a tweet either belongs to "complaint and grievances (C&G)" class or is identified as "unknown". Previous work indicates that the performance of an SVM classifier can be improved by modifying the kernels of the classifier. To investigate the performance of our proposed approach and evaluate the performance across various dimensions, we train our model by varying the kernel parameter of our SVM: linear, polynomial and RBF (Radial Basis Function) Kernels. Further, we use the application of ensemble methods to boost the performance of our baseline SVM classifier by combining kernels into cascaded and parallel manner.

In addition to the identification of complaint reports, we also identify the issue reported in the complaints for a quick addressal by concerned department. Due to the diversity in issues reported in complaints, we perform topic modeling on C&G reports. We use Alchemy Concept API by IBM Watson and identify the hidden topics in complaints. For example, in @RailMinIndia, delay in train, refund-issue, cleanliness, poor service and assistance in train coach and several more related complaints. Similarly, in @dtptraffic, complaints focus on the topics like traffic rules violation, illegal challans, riding motor-bikes without helmets and similar complaints with different issues can be there. We address the challenges of keyword spotting methods and use NLP based methods to find such words and label these complaints into the most likely topic and sub-topic defined in the taxonomy hierarchy. Examples include, riding without helmet or driving without a number plate comes under traffic violation related complaints. More examples: unhygienic food serving or low quality facilities to train passengers are tagged as poor assistance in coaches.

5.2 Case Study 2

Based on our inspection of complaints reported posted to @MORTHIndia and @nitin_gadkari, we divide tweets into 3 categories: Useful tweets, Nearly-Useful tweets, and Irrelevant tweets. We use Rule-based classifier trained on the features extracted in previous phase (problem, location and landmark).

Irrelevant Tweets (IRT): We define a tweet as irrelevant if the post is not about the poor conditions and irregularities of roads or highways causing life risks, discomfort, hazard or poor experience to the citizens. We observe that the tweet labeled as irrelevant is either off-topic or the authors discuss the problems related to road and transport; however the complaint is not about the poor conditions of roads or faulty and dysfunctional facilities.

Useful Tweets (UT): Useful tweets U_t are the posts which are a clear indicator of complaints and can be used to identify the low-level details of the issue faced by the citizens. Given a tweet t_i and a set of named entities $X = < T_c, T_l, T_p >$, we define t_i as a useful tweet- $t_i \in U_t$ if $t_i \in N$ (dataset) and $t_i = \{X, T_o \mid \forall x \in X : P(x)$ where $P(x) \neq \phi\}$. While, T_c denotes the name of city or region, T_l denotes exact geographical location or landmark, T_p denotes the problem or issue reported in the complaint and T_o denotes other words in the tweet.

Nearly-Useful Tweets (N-UT): Nearly-Useful tweets are the tweets posted for complaining a report but containing incomplete or insufficient information about the issue. For example, missing the exact location of the problem, ambiguity in reporting the issue, defining the problem on an abstract level and lacking the details. Given a tweet t_i, we define it as a nearly-useful tweet $t_i \in NU_t$ if $t_i \in N$ and $t_i = \{X, T_o \mid \exists x \in X : P(x)$ where $P(x) = \phi\}$. For simplicity of tweets, we removed the @username mentions from the tweets and used the corrected and enriched form of tweets (after applying micropost-enricment algorithm). We further apply the geographical location hierarchy model (bottom-up) and use graph backtracing method to identify the region or locality for a given pinpoint location in the tweet. For a given pinpoint location T_l, we use OpenStreetMap API and extract the detailed information associated with a geographical location.

6 Empirical Analysis and Results

In this Section, we discuss the empirical analysis performed on the tweets, and acquired experimental results performed on the complaint reports.

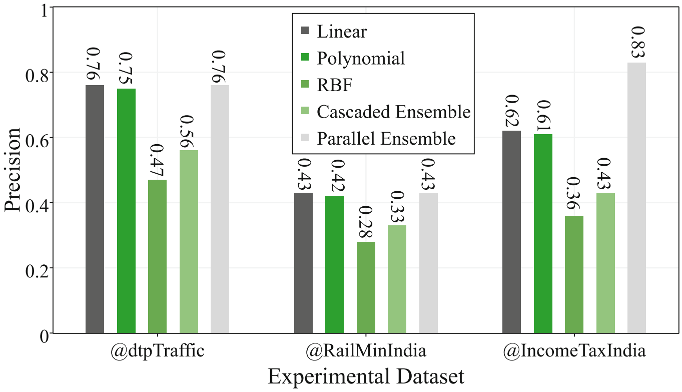


Fig. 1. Confusion matrix for C&G Tweets classifiers. Three different SVM kernel functions & Ensemble Classifiers. Column graphs illustrate - linear kernel outperforms other kernels. Ensembling all kernels in cascaded or parallel boost overall performance of every kernel

6.1 Case Study 1

Proposed AISP classifier identifies 47 (A:12, IS:27, P:8), 132 (A:20, IS: 99, P: 13), 121 (A:41, IS:76, P:4) and 35 (A:4, IS:30, P:1) tweets as AISP for @dtpTraffic, @DelhiPolice, @RailMinIndia and @IncomeTaxIndia respectively. According to our empirical results, we are able to correctly categorize 124, 34, 32 and 103 tweets for @DelhiPolice, @dtpTraffic, @IncomeTaxIndia and @RailMinIndia respectively. Figure 1 reveals that linear kernel in SVM outperforms RBF kernel with a reasonably high margin (variation from 20% to 30%). Figure 1 reveals that for @dtpTraffic (which is linear kernel), we are able to attain the maximum precision rate: 76% $(184/(184 + 58))$. However, for @IncomeTaxIndia & @RailMinIndia, we were able to attain a precision value of 62% $(31/(31+19))$ and 43% $(188/(188+252))$ respectively. Our result shows that using linear kernel in one-class SVM classifier, we are able to get an accuracy upto 60%. However, there is an overall misclassification of upto 10% in detecting complaint tweets as unknown. The column graphs in Fig. 1 reveals that linear and polynomial kernels gives similar results with a small difference 1% to 2% in precision value. Using SVM polynomial kernel, in @dtpTraffic experimental dataset, we were able to identify complaint tweets with a precision of 75% $(170/(170 + 56))$. While, for @RailMinIndia and @IncomeTaxIndia, we were able to identify complaints tweets with a precision rate of 42%$(139/(139 + 189))$ and 61% $(22/(22 + 14))$ respectively. In order to compute the efficacy of our approach for correct classification, we record an overall misclassification of 12% (complaint tweets wrongly classified as unknown) for all accounts for polynomial kernel SVM classifier. Parallel ensemble SVM classifier outperforms cascaded ensemble classifier. Using a combination of linear, polynomial and RBF kernels in parallel manner, we were able to achieve a precision of 75%, 83% and 39% for @dtpTraffic, @IncomeTaxIndia and @RailMinIndia respectively. While, there is an overall misclassification of 18% in identifying complaint tweets as unknown. Figure 1 reveals that by arranging these kernels in cascaded order, it decreases the performance of overall classification from 10% to 20%. For example, for @dtpTraffic and @IncomeTaxIndia datasets, we achieve a precision of 56% and 43% respectively which are approximately 20% lesser than the individual precision of linear kernel SVM classifier. In comparison to cascaded ensembling, in parallel ensemble classification, we are able to boost the accuracy for @IncomeTaxIndia dataset by 21% whereas for @dtpTraffic, the performance is maintained with a precision of 76%.

6.2 Case Study 2

We conduct our experiments on 3, 302 random sampled tweets collected for @MORTHIndia and @nitin_gadkari and report the accuracy of AISP and complaint tweets classifiers. Proposed AISP classifier identifies a total of 20.5% (680 out of 3302) tweets as certainly non-complaint (AISP) reports. Among 680 AISP tweets, 417 tweets are identified as news or information sharing tweets. While, 166 and 97 tweets are identified as appreciation and promotional tweets respectively. Based on our AISP experimental results, we perform rule-based classifier on the remaining 2, 622 tweets and identify useful, nearly-useful and irrelevant complaint reports. Table 6 reveals that a very small percentage of tweets (6.4% - 170 tweets out of 2622 reports) are identified as complete reports that contains all three important components of a killer road complaint. Whereas, the largest chunk of reported tweets is classified as incomplete or nearly-useful tweets (1718 reports out of 2622 tweets). Our experimental results reveal that further only a very small percentage of tweets are convertible (N-UT-C) into complete or useful tweets (50 tweets out of 1718 nearly-useful tweets) while 97% (1668 out of 1718) of nearly-useful tweets have either landmark or concrete problem component missing from the tweets and hence not-convertible.

Table 6. Confusion matrix results for the rule-based classifier

		Predicted			Total
		N-UT	IRT	UT	
Actual	N-UT	1088	131	59	1278
	IRT	376	569	17	962
	UT	254	34	94	382
Total		1718	734	170	**2,622**

Based on the results acquired by our rule-based classifier, we classify bad road related complaints with an overall accuracy of 67%. In addition to measuring the accuracy of our classifier, we also measure the overall recall value of the classification. Based on our results and the Table 6, we record a recall of 65%. The complaints reported to public agencies' accounts are user-generated content and lacks a standard format or terminology for complaining a report. Further, the excessive use of metaphor and sarcasm while reporting a complaint generates false alarms and impacts the overall accuracy of the classification.

7 Conclusions

We present case studies on identification of complaints reported to public agencies' accounts on Twitter. We formulate our problem as a one class classification problem and conduct two case studies on "complaints seeking for an immediate response" and "complaints reported to bring the attention of the government (bad road conditions)". We identify various linguistic and contextual features for identifying complaints reports tweets. We also propose various features that are strong indicators of a tweet to certainly not to be a complaint report. Our results reveal that linear kernel one-class SVM outperforms RBF with a margin of 20% while polynomial and linear kernels produce the similar results with a difference of 1% to 2% of precision. Furthermore, parallel ensembling of kernels outperforms cascaded ensembling. In second case study, we apply rule based classifier on three important components of a killer road complaint; problem reported in the complaint, landmark or pinpoint location, city or location information. Our results shows that the proposed approach classifies these complaint reports with an accuracy of 67% and a recall of 65%. Our results reveal that maximum number of complaints are reported about the risky and accident prone roads while most of them are due to the poor condition of amenities.

References

1. Agarwal, S., Mittal, N., Sureka, A.: Potholes and bad road conditions-mining Twitter to extract information on killer roads. In: The ACM India Joint International Conference (CoDS-COMAD), India. ACM (2018, under-review)
2. Agarwal, S., Sureka, A.: Using KNN and SVM based one-class classifier for detecting online radicalization on Twitter. In: Natarajan, R., Barua, G., Patra, M.R. (eds.) ICDCIT 2015. LNCS, vol. 8956, pp. 431–442. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-14977-6_47
3. Agarwal, S., Sureka, A.: Investigating the potential of aggregated tweets as surrogate data for forecasting civil protests. In: Proceedings of the 3rd IKDD Conference on Data Science, p. 8. ACM (2016)
4. Agarwal, S., Sureka, A., Goyal, V.: Open source social media analytics for intelligence and security informatics applications. In: Kumar, N., Bhatnagar, V. (eds.) BDA 2015. LNCS, vol. 9498, pp. 21–37. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-27057-9_2
5. Anderson, M., Lewis, K., Dedehayir, O.: Diffusion of innovation in the public sector: Twitter adoption by municipal police departments in the US. In: Portland International Conference on Management of Engineering and Technology (2015)
6. D'Andrea, E.: Real-time detection of traffic from Twitter stream analysis. IEEE Trans. Intell. Transp. Syst. **16**(4), 2269–2283 (2015)

7. Edwards, A., de Kool, D.: Webcare in public services: deliver better with less? In: Nepal, S., Paris, C., Georgakopoulos, D. (eds.) Social Media for Government Services, pp. 151–166. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-27237-5_8
8. Frias-Martinez, V., Sae-Tang, A., Frias-Martinez, E.: To call, or to tweet? Understanding 3-1-1 citizen complaint behaviors. In: ASE (2014)
9. Fu, K., Nune, R., Tao, J.X.: Social media data analysis for traffic incident detection and management. In: Transportation Research Board 94th Annual Meeting (2015)
10. Gu, Y., Qian, Z.S., Chen, F.: From Twitter to detector: real-time traffic incident detection using social media data. Transp. Res. Part C Emerg. Technol. **67**, 321–342 (2016)
11. Heverin, T., Zach, L.: Twitter for city police department information sharing. Proc. Am. Soc. Inf. Sci. Technol. **47**, 1–7 (2010)
12. Khan, G.F., Swar, B., Lee, S.K.: Social media risks and benefits a public sector perspective. Soc. Sci. Comput. Rev. **32**(5), 606–627 (2014)
13. Kumar, A., Jiang, M., Fang, Y.: Where not to go? Detecting road hazards using Twitter. In: SIGIR. ACM (2014)
14. Mai, E., Hranac, R.: Twitter interactions as a data source for transportation incidents. In: Proceedings of Transportation Research Board 92nd Annual Meeting (2013)
15. Meijer, A.J., Torenvlied, R.: Social media and the new organization of government communications an empirical analysis of Twitter usage by the Dutch police. Am. Rev. Public Adm. (2014)
16. Mittal, N., Agarwal, S., Sureka, A.: Got a complaint?- Keep calm and tweet it!. In: Li, J., Li, X., Wang, S., Li, J., Sheng, Q.Z. (eds.) ADMA 2016. LNCS (LNAI), vol. 10086, pp. 619–635. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49586-6_44
17. Schulz, A., Ristoski, P., Paulheim, H.: I see a car crash: real-time detection of small scale incidents in microblogs. In: Cimiano, P., Fernández, M., Lopez, V., Schlobach, S., Völker, J. (eds.) ESWC 2013. LNCS, vol. 7955, pp. 22–33. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41242-4_3
18. Wanichayapong, N.: Social-based traffic information extraction and classification. In: 11th International Conference on ITS Telecommunications (ITST). IEEE (2011)

Author Index