Studies in Big Data 8

Witold Pedrycz
Shyi-Ming Chen *Editors*

Information Granularity, Big Data, and Computational Intelligence

Springer

Studies in Big Data

Volume 8

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

For further volumes:
http://www.springer.com/series/11970

About this Series

The series “Studies in Big Data” (SBD) publishes new developments and advances in the various areas of Big Data-quickly and with a high quality. The intent is to cover the theory, research, development, and applications of Big Data, as embedded in the fields of engineering, computer science, physics, economics and life sciences. The books of the series refer to the analysis and understanding of large, complex, and/or distributed data sets generated from recent digital sources coming from sensors or other physical instruments as well as simulations, crowd sourcing, social networks or other internet transactions, such as emails or video click streams and other. The series contains monographs, lecture notes and edited volumes in Big Data spanning the areas of computational intelligence incl. neural networks, evolutionary computation, soft computing, fuzzy systems, as well as artificial intelligence, data mining, modern statistics and Operations research, as well as self-organizing systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

Witold Pedrycz · Shyi-Ming Chen
Editors

Information Granularity, Big Data, and Computational Intelligence

Springer

Editors

Witold Pedrycz
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, AB
Canada

Shyi-Ming Chen
Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei
Taiwan

ISSN 2197-6503 ISSN 2197-6511 (electronic)
ISBN 978-3-319-08253-0 ISBN 978-3-319-08254-7 (eBook)
DOI 10.1007/978-3-319-08254-7
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014944331

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The recent pursuits emerging in big data processing, interpretation, collection, and organization have emerged in numerous sectors including business, industry, and not-for-profit organizations. Data sets such as customer transactions for a mega-retailer, weather monitoring, intelligence gathering can quickly outpace the capacity of traditional techniques and tools of data analysis. We have been witnessing an emergence of new techniques and tools including NoSQL databases, MapReduce, Natural Language Processing, Machine Learning, visualization, acquisition, and serialization.

It becomes imperative to fully become aware what happens when big data grows up: how they are being applied and where they start playing a crucial role. We also need to become fully become aware of implications and requirements imposed on the existing techniques and various methods under development.

Soft Computing regarded as a plethora of technologies of fuzzy sets (or Granular Computing, in general), neurocomputing, and evolutionary optimization brings forward a number of unique features that might be instrumental to the development of concepts and algorithms to deal with big data. In particular, setting up a suitable and fully legitimate level of abstraction by forming semantically meaningful information granules is of paramount relevance. In light of their sheer volume, big data may call for distributed processing, where results of intensive data mining realized locally are afterwards reconciled leading to information granules of higher type. Neurocomputing operating at information granules leads to more tractable learning tasks. Evolutionary computing delivers an essential framework supporting global optimization.

In light of the inherent human-centric facet of Granular Computing the principles and practice of Computational Intelligence have been poised to play a vital role in the analysis, design, and interpretation of the architectures and functioning of mechanisms of big data.

Our ultimate objectives of this edited volume is to provide the reader with an updated, in-depth material on the emerging principles, conceptual underpinnings, algorithms, and practice of Computational Intelligence in the realization of concepts and implementation of big data architectures, analysis, and interpretation as well as data analytics.

An overall concise characterization of the objectives of the edited volume is expressed by highlighting several focal points:

- Systematic exposure of the concepts, design methodology, and detailed algorithms. In general, the volume adheres to the top-down strategy starting with the concepts and motivation and then proceeding with the detailed design that materializes in specific algorithms and representative applications.
- Individual chapters with clearly delineated agenda and well-defined focus and additional reading material available via carefully selected references.
- A wealth of carefully structured and organized illustrative material. The volume includes a series of brief illustrative numeric experiments, detailed schemes, and more advanced problems. They make the material more readable and appealing.
- Self-containment. Given the emerging character of the area of big data, our ultimate intent is to deliver a material that is self-contained and provides the reader with all necessary prerequisites and, if necessary, augments some parts with a step-by-step explanation of more advanced concepts supported by a significant amount of illustrative numeric material and some application scenarios to motivate the reader and make some abstract concepts more tangible.

The area of big data is highly diversified and this volume offers a quite representative view of the area. The contributions published here can be organized into three main parts. The first part, Fundamentals, which comprises chapters "Nearest Neighbor Queries on Big Data" to "Building Fuzzy Robust Regression Model Based on Granularity and Possibility Distribution" is focused on the methodological issues covering a broad spectrum of the approaches and detailed algorithmic pursuits including essential topics of forming cliques in big data, exploiting robust regression and its variants, constructing and optimizing rule-based models, Latent Semantic Indexing, information granulation, and Nearest Neighbor Querying. Part II entitled Architectures consisting of chapters "The Role of Cloud Computing Architectures in Big Data" to "The Web Know ARR Framework: Orchestrating Computational Intelligence with Graph Databases" is aimed at looking at the dedicated computing architectures such as cloud computing and the use of data storage techniques. Part III (case studies) includes chapters "Customer Relationship Management and Big Data Mining" to "Application of Computational Intelligence on Analysis of Air Quality Monitoring Big Data" which offer a suite of studies serving as a testimony to a wealth of promising applications including among others Customer Relationship management, market movements, weather forecasting, and air quality monitoring.

Given the theme of this project, this book is aimed at a broad audience of researchers and practitioners. Owing to the nature of the material being covered and the way it is organized, one can project with high confidence that it will appeal to the well-established communities including those active in various disciplines in which big data, their analysis, and optimization are of genuine relevance. Those involved in data mining, data analysis, management, various branches of engineering, and economics will benefit from the exposure to the subject matter.

Considering a way in which the edited volume is structured, this book could serve as a highly useful reference material for graduate students and senior undergraduate students in courses such as those on intelligent system, data mining, pattern recognition, decision-making, Internet engineering, Computational Intelligence, management, operations research, and knowledge-based systems.

We would like to take this opportunity to express our sincere thanks to the authors for sharing the results of their innovative research and delivering their insights into the area. The reviewers deserve our thanks for their constructive and timely input. We greatly appreciate a continuous support and encouragement coming from the Editor-in-Chief, Prof. Janusz Kacprzyk whose leadership and vision makes this book series a unique vehicle to disseminate the most recent, highly relevant and far-reaching publications in the domain of Computational Intelligence and its various applications.

We hope that the readers will find this volume of genuine interest and the research reported here will help foster further progress in research, education, and numerous practical endeavors.

Witold Pedrycz
Shyi-Ming Chen

Contents

Part I Fundamentals

Part II Architectures

Part III Case Studies

Part I
Fundamentals

Nearest Neighbor Queries on Big Data

Georgios Chatzimilioudis, Andreas Konstantinidis
and Demetrios Zeinalipour-Yazti

Abstract *k Nearest Neighbor (kNN)* search is one of the simplest non-parametric learning approaches, mainly used for classification and regression. *kNN* identifies the *k* nearest neighbors to a given node given a distance metric. A new challenging *kNN* task is to identify the *k* nearest neighbors for all nodes simultaneously; also known as *All kNN (AkNN)* search. Similarly, the *Continuous All kNN (CAkNN)* search answers an *AkNN* search in real-time on streaming data. Although such techniques find immediate application in computational intelligence tasks, among others, they have not been efficiently optimized to this date. We study specialized scalable solutions for *AkNN* and *CAkNN* processing as demanded by the volume–velocity-variety of data in the Big Data era. We present an algorithm, coined *Proximity*, which does not require any additional infrastructure or specialized hardware, and its efficiency is mainly attributed to our smart search space sharing technique. Its implementation is based on a novel data structure, coined k^+-heap. *Proximity*, being parameter-free, performs efficiently in the face of high velocity and skewed data. In our analytical studies, we found that *Proximity* provides better time complexity compared to existing approaches and is very well suited for large scale scenarios.

Keywords k Nearest neighbors · Big data · Computational intelligence · Smartphones

G. Chatzimilioudis (✉) · A. Konstantinidis · D. Zeinalipour-Yazti
Department of Computer Science, University of Cyprus, 1 University Avenue,
P.O. Box 20537, 1678 Nicosia, Cyprus
e-mail: gchatzim@cs.ucy.ac.cy

A. Konstantinidis
e-mail: akonstan@cs.ucy.ac.cy

D. Zeinalipour-Yazti
e-mail: dzeina@cs.ucy.ac.cy

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_1,

1 Introduction

The k Nearest Neighbor (kNN) search [1] is one of the simplest non-parametric learning approaches, mainly used for classification [2] and regression [3]. The kNN of an object o_a from some dataset O, denoted as $kNN(o_a, O)$, are the k objects whose attributes are most similar to the attributes of o_a [1]. Formally, $\forall o_b \in kNN(o_a, O)$ and $\forall o_c \in O - kNN(o_a, O)$ given $o_a \neq o_b \neq o_c$, it always holds that $dist(o_a, o_b) \leq dist(o_a, o_c)$, where $dist$ can be any L_p-norm metric, such as Manhattan (L_1), Euclidean (L_2) or Chebyshev (L_∞).

kNN search is a classical Computational Intelligence problem with extensions that include the Condensed nearest neighbor (CNN, the Hart algorithm) algorithm that reduces the data set for kNN classification [4] and the fuzzy-kNN [5] that deals with uneven and dense training datasets. kNN further finds applicabilities in several domains such as computational geometry [6–8], image processing [9, 10], spatial databases [11, 12], and recently in social networks [13].

A new challenging *kNN* task is to identify the *k* nearest neighbors for all nodes simultaneously; also known as *All kNN* (*AkNN*) search. An AkNN search is viewed as a generalization of the basic kNN search that computes the $kNN(o, O) \forall o \in O$. In temporal and streaming data a similar task of interest is the *Continuous All kNN* (*CAkNN*) search, which answers an AkNN query in real-time.

AkNN and CAkNN are new and challenging computational intelligence problems, which cannot be efficiently tackled using existing techniques. Thus, they may serve as both a real-world benchmark and an improvement technique of computational intelligence methods. For example, CAkNN can be used for both real-time classification and regression in Big Data cases where the volume-velocity-variety of data is high and cannot be handled by conventional techniques. Consider classifying tweets provided by Twitter users in real time (with 100 billion daily active users and 143,199 tweets per second in 2013). Furthermore, it can be combined with Neural Network [14] to improve its real-time predictability when new data arrive again with high velocity-volume-variety, it can also be used with minor modifications to extend the well-known Variable Neighborhood search approach [15] for tackling combinatorial and global optimization problems. Finally, AkNN and CAkNN can be combined with Multi-Objective Evolutionary Computation approaches [13] for improving their performance in terms of speed and efficiency when dealing with Multi-objective Optimization Problems (MOPs). For example, it can be combined with MOEA based on Decomposition (MOEA/D) [16] for finding neighbors of each solution in the weight space faster, or it can be combined with the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [17] for improving the crowding-distance estimation (i.e., a techniques utilized for improving the diversity of the final result).

The advances in technology and the automatization of many processes in numerous sectors, including business, industry, and not-for-profit organizations, has led to the pursuit of big data processing, interpretation, collection and organization. For example, the proliferation of smart devices and sensors with the trend to share, communicate and store data has brought an information explosion in spatio-temporal applications, with ever increasing amounts of data that need to be efficiently managed. The CAkNN task is of great interest since it offers a new dimension of neighborhood “sensing”. Applications of this neighborhood “sensing” capability could enhance public emergency services like E9-1-1 [18] and NG9-1-1 [19], and facilitate the uptake of location-based social networks (e.g., Rayzit [20], Waze [21]).

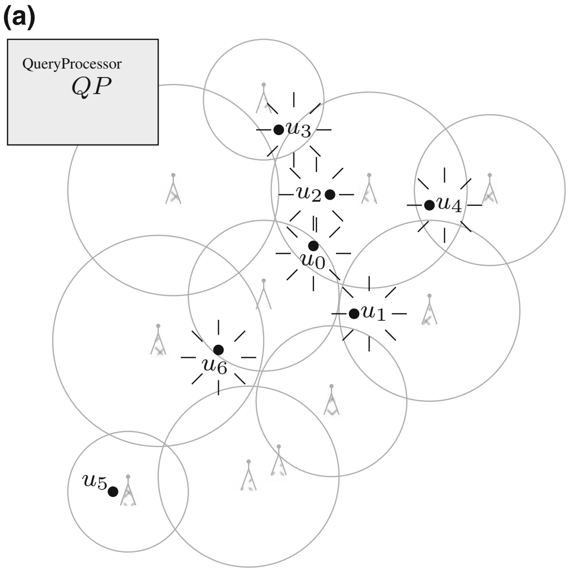
In this chapter, we study the problem of efficiently processing a CAkNN search in a cellular or WiFi network, both of which are ubiquitous. We present an algorithm, coined *Proximity*, which does not require any additional infrastructure or specialized hardware and its efficiency is mainly attributed to a smart *search space sharing* technique we present and analyze. Its implementation is based on a novel data structure, coined k^+-heap. *Proximity*, being parameter-free, performs efficiently in the face of high mobility and skewed distribution of users (e.g., the service works equally well in downtown, suburban, or rural areas).

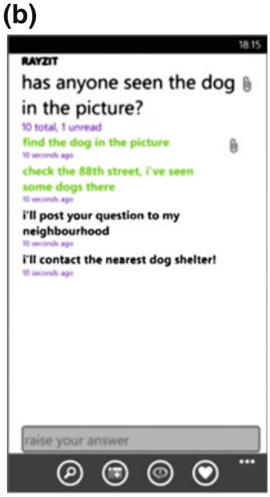
Consider a set of smartphone users moving in the plane of a geographic region. Let such an area be covered by a set of *Network Connectivity Points* (*NCP*) (e.g., cellular towers of cellular networks, WiFi access points of wireless 802.11 networks etc.) Each *NCP* inherently creates the notion of a *cell*. Without loss of generality, let the cell be represented by a circular area[1] with an arbitrary radius. A mobile user u is serviced at any given time point by one *NCP*, but is also aware of the other *NCP*s in the vicinity whose communication range reach u (e.g., cell-ids of different providers in an area, or MAC addresses of WiFi hot-spots in an area).

To illustrate our abstraction, consider the example network shown in Fig. 1, where we provide a micro-blogging chat channel [20] between each user u and its $k = 2$ nearest neighbors. In the given scenario, each user concurrently requires a different answer-set to a globally executed query, as shown in the caption of Fig. 1. Notice that the answer-set for each user u is not limited within its own *NCP* and that each *NCP* has its own communication range. Additionally, there might be areas with dense user population and others with sparse user population. Consequently, finding the k-nearest neighbors of some arbitrary user u could naively involve from a simple lookup in the *NCP* of u to a complex iterative deepening into neighboring *NCP*s, as we will show in Fig. 3b.

The remaining of this chapter is organized as follows: Sect. 2 introduces fields for which the proposed framework caters for. Section 3 defines our system model and the problem. Section 4 provides the related work necessary for understanding the foundations of this work. Section 5 presents the *Proximity* framework and a breakdown of our data structures and algorithms. Section 6 finally summarizes and concludes the knowledge acquired from existing research and discusses our future plans.

[1] Using other geometric shapes (e.g., hexagons, Voronoi polygons, grid-rectangles, etc.) for space partitioning is outside the scope of this paper.





RAYZIT

has anyone seen the dog in the picture?

10 total, 1 unread

find the dog in the picture

10 seconds ago

check the 88th street, i've seen some dogs there

10 seconds ago

i'll post your question to my neighbourhood

10 seconds ago

i'll contact the nearest dog shelter!

10 seconds ago

raise your answer

Fig. 1 **a** A snapshot of a cellular network instance, where the 2-nearest neighbors for u_0 are $\{u_1, u_2\}$. Similarly for the other users: $u_1 \rightarrow \{u_0, u_2\}, u_2 \rightarrow \{u_3, u_0\}$, $u_3 \rightarrow \{u_2, u_0\}$, $u_4 \rightarrow \{u_2, u_3\}$, $u_6 \rightarrow \{u_0, u_1\}$. **b** Rayzit [20], an example application of a proximity-based micro-blogging chat

2 Background

Big data refers to data sets whose size and structure strains the ability of commonly used relational DBMSs to capture, manage, and process the data within a tolerable elapsed time [22]. The *volume–velocity-variety* of information in this kind of datasets give rise to the big data challenge, which is also known as the 3V challenge.

The *volume* of such datasets is in the order of few terabytes (TB) to petabytes (PB) that are often of high *information granularity*. Examples of such volumes are the U.S. Library of Congress that in April 2011 had more than 235 TB of data stored and the World of Warcraft online game using 1.3 PB of storage to maintain its game, the German Climate Computing Center (DKRZ) storing 60 PB of climate data.

The *velocity* of information in social media applications (such as photovoltaic, traffic and other monitoring apps) can grow exponentially as users join the community. Such growth can produce unprecedented volumes of data streams. For example, Ontario's Meter Data Management and Repository (MDM/R) [23] stores, processes and manages data from 4.6 million smart meters in Ontario, Canada and provides hourly billing quantity and extensive reports counting 110 million meter reads per day on an annual basis that exceeds the number of debit card transactions processed in Canada.

Furthermore, the *variety* of data can be anything from structured (relational or tabular) to semi-structured (XML or JSON) or even unstructured (Web text and log files) data and combination thereof. For example, Google's experimental robot cars [24], which have navigated thousands of miles of California roads, use an artificial-intelligence technique tackling big data challenges, parsing vast quantities of data and making decisions instantaneously.

Due to the high demand for big-data management, the literature witnessed an emergence of new techniques and tools for taming big data. For example, new data management related mechanisms are proposed [25, 26] that exploit the MapReduce framework [27] for analyzing data on racks of servers and NoSQL databases.

Computational Intelligence techniques [28–30] such as fuzzy logic, evolutionary computation, neurocomputing and other machine learning techniques provide us with searching and reasoning means to bring forward solutions to the big data challenges. Information Granularity [31] is a research field that sheds light into processing complex information entities that can be abstracted or detailed in different levels. Information granularity techniques are first-class candidates for breaking volume barriers when it comes to processing big data, focusing the computational power directly to the various levels of abstraction needed. In addition, Evolutionary Computation (EC) techniques are mainly used for optimization tasks, such as finding the parameters/models that optimize some prespecified evaluation criteria given some observed data and can be utilized for dealing with both single and Multi-objective Optimization Problems (MOPs) [13].

As we enter the age of big data, many different evolutionary computation and machine learning techniques have been modified, combined, extended and investigated for their ability to extract insights in an actionable manner. In [32], Stanford and Google researchers developed a deep learning based approach to build an online face detector by training a model on a large dataset of Youtube images (a model with 1 billion connections and a dataset with 10 million 200 × 200 pixel images downloaded from the Internet). The network is trained using model parallelism and asynchronous SGD on a cluster with 1,000 machines (16,000 crores) for 3 days. Furthermore, Hall et al. [33] introduced a Decision tree approach for analyzing big data, which was extended in [34] by combining the Decision Trees with GAs for further improving their performance. Moreover in [35], Lu and Fahn proposed a hierarchical artificial neural network for recognizing high similar large data sets.

3 System Model and Problem Formulation

This section formalizes our system model and defines the problem. The main symbols are summarized in Table 1.

Let U denote a set of smartphone users moving in the plane of a geographic region. Let such an area be covered by a set of *Network Connectivity Points* (*NCP*) (e.g., cellular towers found in cellular networks, WiFi access points found in

Table 1 Notation used throughout this work

Notation	Description
NCP	Network connectivity point
c, C	Single NCP, set of all NCPs
$radius_c$	Range of NCP c
λ	The maximum number of users an NCP can serve
u, U	A single user, set of all users in the network
n	Number of users in the network ($\lvert U \rvert$)
U_c	Set of users of NCP c
r, R	A single user report, all user reports for a single timestep
$loc(u)$	Location of user u
$ncp(u)$	The NCP that a user is registered to
$ncp_{vic}(u)$	List of NCPs whose range cover user u
S_c	The search space of NCP c
d_c	Distance of kth nearest user to the border of NCP c
$kNN(u)$	The set of k-nearest neighbors of user u
$k\text{th}_c$	The kth nearest outside user to the boundary of cell c

wireless networks etc.) Each NCP inherently creates the notion of a *cell*, defined as c_i. Without loss of generality, let the cell be represented by a circular area with radius $radius_c$. The number of users λ serviced by an NCP is a network parameter (cell capacity). A mobile user u is serviced at any given time point by one NCP, but is also aware of the other NCPs that are in its vicinity and whose communication range cover it (e.g., cell-ids of different providers in an area, or MAC addresses of WiFi hot-spots in an area, etc).

Assume that there is some centralized (or cloud-like) service, denoted as QP (Query Processor), which is accessible by all users in user set U. Allow each user u to report its positional information to QP regularly. These updates have the form $r_u = \{u,\ loc(u),\ ncp(u),\ ncp_{vic}(u)\}$, where $loc(u)$ is the location of user u,[2] $ncp(u)$ is the NCP user u is registered to and $ncp_{vic}(u)$ is a list of NCPs in the vicinity of u.

The problem we consider in this work is how to efficiently compute the k-nearest neighbors of all smartphones that are connected to the network, at all times. We consider a *timestep* that defines rounds where we need to recompute the kNNs of the users. Depending on the application, this can take place either at a preset time interval or whenever we have a number of new user location updates arriving at the server. Formally, we aim to solve a problem we coin the $CAkNN$ problem.

Definition 1 (*CAkNN problem*) Given a set U of n points in space and their location reports $r_{i,t} \in R$ at *timestep* $t \in T$, then for each object $u_i \in U$ and *timestep* $t \in T$, the $CAkNN$ problem is to find the k objects $U_{sol} \subseteq U - u_i$ such that for all other objects $u_o \in U - U_{sol} - u_i,\ dist(u_k,\ u_i) \leq dist(u_o,\ u_i)$ holds.

[2] The location of a user can be determined either by fine-grain means (e.g., AGPS) or by coarse-grain means (e.g., fingerprint-based geo-location [36]).

In order to better illustrate our definition, consider Fig. 2, where we plot a *timestep* snapshot of 7 users $u_0 - u_6$ moving in an arbitrary geographic region. The result for this *timestep* to a $k = 2$ query would be $kNN(u_0) = \{u_1, u_2\}$, $kNN(u_1) = \{u_0, u_2\}$, $kNN(u_2) = \{u_3, u_0\}$, $kNN(u_3) = \{u_2, u_0\}$, $kNN(u_4) = \{u_2, u_0\}$, $kNN(u_6) = \{u_7, u_1\}$.

Obviously, the solution for a user u will not always reside inside the same *NCP* cell c, but might reside in neighboring cells or even further (e.g., if neighboring cells do not have any users). Computing a separate search space for every user is very expensive. On the other hand, *search space sharing* is achieved when the same search space is used by multiple users and it guarantees the correct *kNN* solution for all of them. If we apply this reasoning for all users U_c in c, then the common search space S_c for U_c would be defined as the union of the individual search spaces of every user in U_c. We efficiently build S_c with the assistance of complementary data structures we devise in this work and explain next. In Fig. 2, the search space constructed by our framework for users u_0 and u_6 is the largest dotted circle.

4 Related Work

In this section we provide an extensive coverage of the k nearest neighbor related work tackling static data, continuous data and distributed data.

4.1 KNN Queries on Static Data

For applications where data is represented by a linear array, constant time algorithms have been proposed to solve the *All Nearest Neighbor* (*ANN*) and *All k-Nearest Neighbor* (*AkNN*) problems. There has been extensive work in the field of image processing and computational geometry (e.g., [9, 10]).

In Euclidean space (and general metric spaces), there has been also extensive work on solving the *ANN* and *AkNN* problems. For large datasets residing on disk (external memory), works like Zhang et al. [12] and Chen et al. [11] exploit possible indices on the datasets and propose algorithms for R-tree based nearest neighbor search.

For small ANN and AkNN problems in Euclidean space, where data fits inside main memory, early work in the domain of computational geometry has proposed solutions. Clarkson et al. [7] was the first to solve the ANN problem followed by Gabow et al. [8], Vaidya [37] and Callahan [6]. Given a set of points [7, 8] use a special quad-tree and [37] use a hierarchy of boxes to divide the data and compute the ANN. The worst case running time, for both building the needed data structures and searching in these techniques, is $O(nlogn)$, where n is the number of points in the system. For the AkNN problem works [7] propose an algorithm with $O(kn + nlogn)$ and [37] an algorithm with $O(knlogn)$ time complexity.

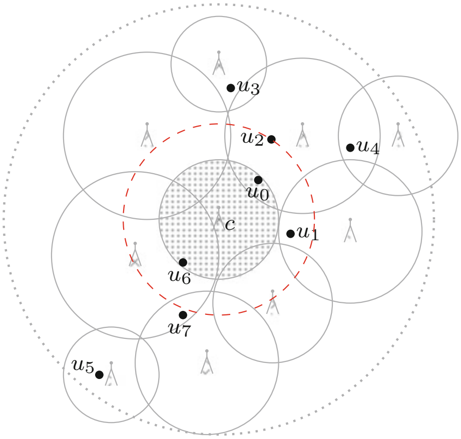


Fig. 2 The *search space* of cell *c* is the big circle with the *dotted outline*. Any user inside this circle is a *kNN* candidate for any user inside *c*

For multi-dimensional disk-resident data kNN Joins have been optimized in [11, 12, 28, 38–40]. Zhang et al. [12] index the disk-resident data using an R-tree and develop an efficient depth-first traversal algorithm and a hash based algorithm. In [11] the authors propose the minimum bounding rectangle enhanced quad-tree as well as a new distance measure for pruning as many distance comparisons as possible. In [28], a specialized index together with an optimal page loading strategy were proposed to reduce both CPU and I/O cost for disk-resident data. Xia et al. [38] optimizes kNN Joins for high dimensional data by hashing points into blocks and sorting the blocks for a nested loop join. Xia and Yao [39, 40] similarly propose efficient indexing techniques for avoiding scanning the whole dataset repeatedly and for pruning as many distance computations as possible.

4.2 Continuous KNN Queries

When it comes to streaming updates of object attributes, main memory processing is usually mandatory for spatio-temporal applications, where objects are highly mobile.

Yu et al. [41] followed by Mouratidis et al. [42] optimize Continuous kNN queries. Objects are indexed by a grid in main memory given a system-defined parameter value for the grid size. For each query they both use a form of iteratively enlarging a range search to find the kNN. For small object speeds and/or low object agility, both propose a *stateful* technique to incrementally compute the result of a query of the current timestep using the result of the previous timestep. They define an influence region for the query inside the grid and depending on what happens in this region, the new result is computed using the previous result, minimizing the search space. Whenever the query object moves or the agility and speed of the objects is high, both fall back to their slower *stateless* version where at each timestep the result of the query is computed from scratch.

Chatzimilioudis et al. [43] is the only work that optimizes Continuous All-kNN queries, termed Continuous-AkNN, in a centralized environment. The core intuition behind this work is geographically partitioning the object space based on the transmission radius of the network connectivity points (e.g., WiFi router, cellular base-station, etc.) and determining a candidate set for each network connectivity point. Then, each object υ only scans the candidate set of its connectivity point and determines its kNN.

4.3 All-KNN Queries in Distributed Systems

The high information granularity of the location updates is very restrictive for disk-based storage and indexing. Therefore, distributed and parallel techniques for memory-resident data is desirable and demands optimization in respect to the CPU time.

Callahan's [6] main contribution is a parallel algorithm that solves the AkNN problem in $O(logn)$ using $O(n)$ processors with *shared-memory*. Given a set of points, Callahan use a special quad-tree to divide the data. His algorithm has $O(kn + nlogn)$ time complexity.

Recently, solutions utilizing large-scale distributed data processing have been proposed by Lu et al. [25] optimized for exact kNN Joins and Zhang et al. [26] optimized for approximate kNN Joins.

Zhang et al. [26] give a MapReduce technique that optimizes AkNN queries. It splits A into equally-sized $\sqrt{m}$ disjoint subsets, i.e., $A = \bigcup_{1 \leq l \leq \sqrt{m}} A_l$, creating m distinct combinations of 2 subsets $\{A_l, A_h\}$. In a first parallel phase, each server is assigned one such pair of subsets and scans A_h to find the kNNs for each object in A_l. All intermediate answers are then collected and distributed in a second parallel phase to compute the top-k neighbors for each object in A.

Lu et al. [25] propose a 2-job Map-Reduce solution for optimizing AkNN queries on static datasets. In a centralized pre-processing step a set of optimal pivot points are carefully selected. In the first Map-Reduce job, the mappers split set A into m disjoint subsets A_i based on the Voronoi cells generated by the selected pivots, and record the maximum and minimum distance between each pivot and its subset. Each area A_i and corresponding set O_i is mapped to a server s_i. There is no reduce step in their first Map-Reduce job. In the second Map-Reduce job, for each A_i a candidate set $CS_i \subseteq A$ is computed that guarantees to include all kNNs for the subset A_i. Each A_i is mapped to its candidate set CS_i. The reducer s_i then computes the kNNs for each point $o \in O_i$ using CS_i.

4.4 Shortcomings of Existing Work

Techniques that optimize disk I/O are unattractive for solving *CAkNN* queries, since the CPU latency is the actual bottleneck as shown by Chen et al. [11].

Moreover, tree-based techniques proposed for *ANN* queries require super-linear time for their structure build-up phase (as [6–8, 37]) and need to be updated or re-built in every timestep, which is inefficient.

No previous work tackles the problem of continuous all *k*-nearest neighbor (*CAkNN*) queries specifically. In smartphone network applications the users are highly mobile with hard-to-predict mobility patterns and their location distribution is far from uniform [44]. This makes *stateful* techniques inefficient as shown in [41, 42], since keeping previous answers (states) of the query becomes more of a burden than a help for faster query evaluation. Furthermore, in proximity applications considered in this paper, smartphone users are moving and are both the objects of interest and the focal points of queries.

Our framework, *Proximity*, is main-memory based and *stateless*, i.e., no previous data/calculation of the previous evaluation round is used in the current round. A *stateless CAkNN* solution would solve an *AkNN* problem at each timestep. In [43], we compare *Proximity* analytically to the early work of computational geometry [6–8, 37] and show that the running time complexity of our framework is better [i.e., $O(n(k + \lambda))$ as opposed to $O(knlogn)$]. We compare *Proximity* experimentally against an adaptation of state-of-the-art *CkNN* solution [41, 42]. Due to the agility of the realistic mobile datasets used, these works can only make use of their *stateless* algorithm, which solves a *kNN* query in every timestep. Thus, such adaptations can only optimize a *kNN* query for each timestep separately and for each user separately, building a new search space for each user (see Fig. 3). We show that our specialized *Proximity* framework performs better, mainly due the batch processing capability of the *AkNN* queries. The most significant difference is that the *Proximity* framework groups users of the same cell together and uses the same search space for each group (*search space sharing* Fig. 3a).

5 The Proximity Framework

We start out outlining of the *Proximity* framework and the intuition behind its operation. We then describe in detail how the search space is built-up using our k^+-heap data structure and its associated insertion and update algorithms.

5.1 Outline of Operation

The *Proximity* framework is designed in such a way that it is: (i) *Stateless*, in order to cope with transient user populations and high mobility patterns, which complicate the retrieval of the continuous *kNN* answer-set. In particular, we solve the *CAkNN* problem for every timestep separately without using any previous computation or data; (ii) *Parameter-free*, in order to be invariant to parameters that are network-specific (such as cell size, capacity, etc.) and specific to the user-

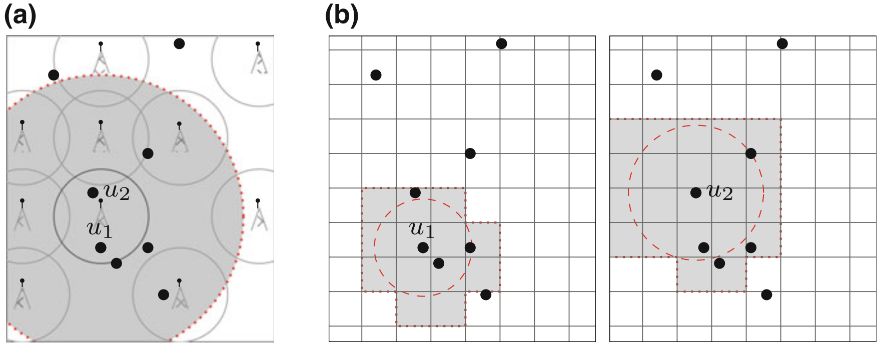


Fig. 3 **a** In *proximity* the search space is pre-constructed for all users of the same cell (e.g., u_1 and u_2); whereas **b** for existing state-of-the-art algorithms the search space needs to be iteratively discovered by expanding a ring search for each user separately into neighboring cells

distribution; (iii) *Memory-resident*, since the dynamic nature of mobile user makes disk resident processing prohibitive; (iv) *Specially designed* for *highly mobile* and *skewed distribution* environments performing equally well in downtown, suburban, or rural areas; (v) *Fast and scalable*, in order to allow massive deployment; and (vi) *Infrastructure-ready* since it does not require any additional infrastructure or specialized hardware.

Algorithm 1 . Proximity Outline

```
Input: User Reports $R$ (single timestep), set $C$ of all NCPs
Output: kNN answer-set for each user in $U$
 1: for all $c \in C$ do
 2:     initialize $k^{+}{}_c$                              ▷ Initialize our $k^{+}$-heap
 3: end for
 4: for all $r \in R$ do                                    ▷ Phase 1: build $k^{+}$-heap
 5:     for all $c \in C$ do
 6:         $insert(r, k^{+}{}_c)$
 7:     end for
 8: end for
 9: $U \leftarrow users(R)$
10: for all $u \in U$ do                                    ▷ Phase 2: scan $k^{+}$-heap
11:     $kNN_u = \emptyset$                                  ▷ Conventional k-max heap
12:     $c \leftarrow r_u.ncp$
13:     for all $v \in k^{+}{}_c$ do
14:         if $v$ is a kNN of $u$ then
15:             $update(kNN_u, v)$
16:         end if
17:     end for
18:     report kNN to node $u$
19: end for
```

For every timestep *Proximity* works in two phases (Algorithm 1): In the first phase one k^{+}-heap data structure is constructed per *NCP*, using the location reports of the users (lines 1–8). In the second phase, the *k*-nearest neighbors for each user are determined by scanning the respective k^{+}-heap and the results are reported back to the users (lines 9–19).

At each timestep the server *QP* initializes our k^{+}-heap for every *NCP* in the network. The k^{+}-heap integrates three individual sub-structures that we will explain next. The user location reports are gathered and inserted into the k^{+}-heap of every *NCP*. After all location reports have been received and inserted, each *NCP* has its search space stored inside its associated k^{+}-heap. After the build phase, each user scans the k^{+}-heap of its *NCP* to find its *k*-nearest neighbors.

5.2 Constructing the Search Space

Here we describe the intuition behind our search space sharing concept. Every user covered by an *NCP* uses the same search space to identify its *kNN* answer-set.

In order to construct a correct search space for each *NCP*, we need to be able to identify nodes that might be part of the *kNN* answer-set for any arbitrary user of a given *NCP*. For instance, consider two users u_0 and u_6, in Fig. 2, which are positioned on the perimeter of their *NCP* c. Also, consider user u_2 being outside c and close to u_0. In such a scenario, the search space for c must obviously include u_2, as it is a better *kNN* candidate to u_0 than u_6. However, even if we were aware of the k closest users to c (besides the users in c), would not allow us to correctly determine the *kNN* for any arbitrary user in c. To understand this, consider again Fig. 2 with a *2NN* query. u_1 and u_2 are the two closest outside nodes to the border of c. Yet, we can visually determine that u_7 is a more appropriate *2NN* candidate for u_6 than all aforementioned nodes, i.e., u_0, u_1, u_2.

To overcome this limitation, we define a prune-off threshold, denoted as $k\text{th}_c$, which determines the size of the search space of c. $k\text{th}_c$ is the kth closest outside user to the border of c, which determines the width d_c of the *search expansion* (striped ring as seen in Fig. 4). Inside this ring there are k users by definition. These k users form the K-set. In our running example $k\text{th}_c = u_2$. This guarantees that the search space will have at least k users. All users at distance less that $2 * radius_c + d_c$ from c's border, are also part of the search space. This guarantees that each user inside c will find its actual kNN inside the search space.

The size of each *NCP* search space depends on the communication area of the *NCP* and the kth closest outside user to the border of its communication area. The users inside c comprise set U_c and the users that are at distance greater than d_c and less than $2 * radius_c + d_c$ from the cell's border comprise set B_c (grey ring in Fig. 4). Set K, set B, and the users U_c inside c form the search space S_c of c.

Definition 2 (*K-set*) Given a set of users $u \in U - U_c$ outside *NCP* cell c that is ordered with ascending distance $dist(u, c)$ to the border of c, set K_c consists of the first k elements of this set (striped ring in Fig. 4).

Definition 3 (*kth outside neighbor of the NCP cell*) Given K_c (ordered as in Definition 2), the kth user is called the Kth nearest neighbor of c and denoted $k\text{th}_c$.

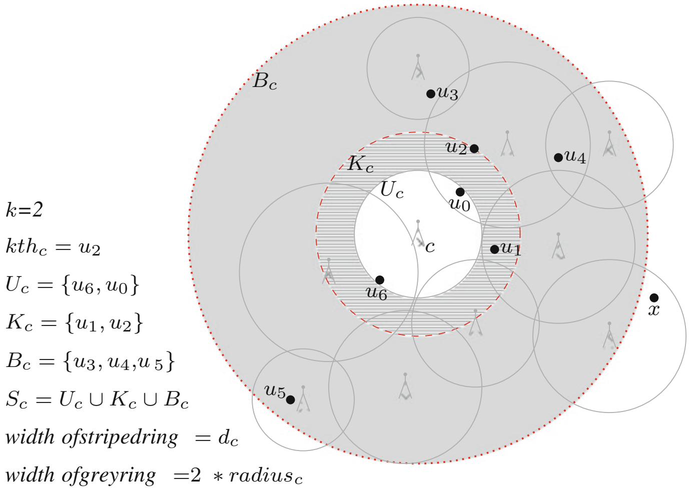

k=2
$kth_c = u_2$
$U_c = \{u_6, u_0\}$
$K_c = \{u_1, u_2\}$
$B_c = \{u_3, u_4, u_5\}$
$S_c = U_c \cup K_c \cup B_c$
width ofstripedring $= d_c$
width ofgreyring $=2 * radius_c$
B_c
u_3
u_2
u_4
K_c
U_c
u_0
c
u_1
u_6
x
u_5

Fig. 4 An example of the common search space for the users inside cell c (*white circle*) for $k = 2$. The search space S_c of c is $\{u_0, u_1, u_2, u_3, u_4, u_5, u_6\}$ and is represented by the *big circle* with the *dotted outline*. Set S_c includes all users inside c (set U_c), the striped ring (set K_c) and the grey ring (set B_c). Any node outside S_c (e.g., user x) is guaranteed NOT to be a *kNN* of any user inside cell c. The 2-nearest neighbors for the nodes in c are $kNN(u_0) = \{u_1,\ u_2\}$ and $kNN(u_6) = \{u_0,\ u_1\}$

Definition 4 (B, *Boundary Set*) Given an *NCP* denoted as c and its kth outside neighbor $k\text{th}_c$, set B_c consists of all users $u \in U - (U_c \cup K)$ with distance $dist(u, c) < dist(k\text{th}_c, c) + 2 * radius_c$ from the border of c. In other words B_c consists of all users $u \in U$ with distance $dist(k\text{th}_c, c) < dist(u, c) < dist(k\text{th}_c, c)$ $+2 * radius_c$.

Definition 5 (S, *Search Space set*) Given an *NCP* c and its K_c set, the search space S_c of c consists of all users $u \in U_c \cup K_c \cup B_c$ (big circle with dotted outline in Fig. 4).

In our Fig. 4 example, at the end of the build phase, the k^+-heap of c includes users $\{u_6, u_0, u_1, u_2, u_3, u_4, u_5\}$. This is the common search space S_c for all users $U_c = \{u_0, u_6\}$ of c, which guarantees to include their exact k-nearest neighbors.

5.3 Specialized Heap: The K^+-Heap

Computing the search space for each cell inefficiently might be prohibitive for the application scenarios we envision as detailed in the introduction. In this section, we show in detail how the search space for an *NCP* is constructed using our k^+-heap data structure. Recall that as user reports arrive at the server *QP* they are inserted into each k^+-heap. A user report either stays inside a k^+-heap or eventually gets evicted using a policy that we will describe later. After all user reports have been probed through the k^+-heap of every *NCP*, each k^+-heap contains the actual search space of its *NCP*. Consequently, the build phase takes a total of $n * |C|$ insertions.

Algorithm 2 . k^+-heap: Insert(u_{new})

```
Input: $u_{new}$, $c$ of $u_{new}$
Output: $k^+{}_c$
 1: $kth_c \leftarrow head(K_c)$
 2: if $dist(u_{new}, c) < radius_c$ then
 3:     $insert(u_{new}, U_c)$
 4: else if $dist(u_{new}, c) < dist(kth_c, c)$ then
 5:     $insert(u_{new}, K_c)$
 6:     if $K$ heap has more than $k$ elements then
 7:         $kth_c \leftarrow pophead(K_c)$
 8:         $insert(kth_c, B_c)$
 9:         $Update\_boundary(head(K_c))$
10:     end if
11: else if $dist(u_{new}, c) < dist(kth_c, c) + 2 * radius_c$ then
12:     $insert(u_{new}, B_c)$
13: else
14:     discard $u_{new}$
15: end if
```

The k^+-heap consists of three separate data structures: a heap for the set K_c and two lists for *Boundary* set B_c and the set U_c. The heap used for set K_c is a conventional k-max-heap. It stores only the k users outside c with the minimum distance $dist(u, c)$ from the border of c. Thus, the heap K has always $k\text{th}_c$ at its head. The *boundary* list is a list ordered by $dist(u, c)$, which stores set B. Its elements are defined by $k\text{th}_c$ (see Definition 4). Similarly, we use a list to store the users $U_c \subseteq U$ of c. Notice that some *NCP* cells will be overlapping, so there are areas where users are inside multiple cells. Such users are inserted into all lists U_j of $c_j \in C$ that cover them. The k^+-heap has $O(1)$ lookup time for the kth nearest neighbor of c. It has worst case $O(\log(k * |B|))$ insertion time and contains $|S_c| = k + |B_c| + |U_c|$ elements.

5.4 Insertion into the K$^+$*-Heap (Algorithm 2 and 3)*

When inserting a new element u_{new} into the k^+-heap of c, we distinguish among four cases (see Algorithm 2): (i) u_{new} is covered by c and belongs to set U_c (line 2), (ii) u_{new} belongs to set K_c (line 4), (iii) u_{new} belongs to set B_c (line 11), or (iv) u_{new} does not belong to the search space $S_c = U_c \cup K_c \cup B_c$ of *NCP* c (line 13). In case (i) the element is inserted into the U_c list. In case (ii) we need to insert u_{new} into heap K (line 5) and move the current head $k\text{th}_c$ from K to the boundary list B (lines 7–8). This yields a new head $k\text{th}'_c$ in K (line 9). Every time the $k\text{th}_c$ changes, the boundary list B needs to be updated, since it might need to evict some elements according to Definition 4. In case (iii) we insert u_{new} into the ordered boundary list

B (line 12). Note that the sets K_c and B_c are formed as elements are inserted into the k^+-heap. The first *k* elements inserted in the empty k^+-heap define the K_c set. In case (iv) the element is discarded.

Algorithm 3 . k^+-heap: Update_boundary(kth_c)

```
Input: $kth_c$ (the $k^{th}$ outside neighbor of NCP $c$)
Output: $B_c$ updated
1: $d \leftarrow dist(kth_c, c) + 2 * radius_c$
2: $i \leftarrow$ element with the max. distance smaller than $d$ using binary search
3: $remove(B_c, i + 1, end)$
```

5.5 *Running Example*

Using Fig. 4 as our network example in timestep *t* we will next present the *Proximity* framework step-by-step.

Server *QP* initiates a k^+-heap for every *NCP* in *C*. The k^+-heap consists of heap *K*, ordered list *B*, and list *U*. The reports that arrive at *QP* are $R = r_0,\ r_1,\ r_2,\ r_3,\ r_4,\ r_5,\ r_6,\ r_x$. Every report is inserted into every k^+-heap on the *QP* (see Algorithm 1, lines 1–5). The order in which the reports are inserted into a k^+-heap does not affect the correctness of the search space.

For our example, assume that the reports are inserted in the order seen in the first column of Table 2. For every insertion we can see the contents of k_c^+ in the same Table. For simplicity we will only follow the operation for the k^+-heap of *NCP c*. When report r_4 is inserted into k_c^+ it ends up inside heap K_c, since user u_4 is not inside *NCP* cell *c* (condition line 2) and heap K_c is empty. Next, report r_x is inserted into k_c^+ and it also ends up inside heap K_c since this is not full yet. When r_2 is inserted, it ends up inside heap K_c (line 5) and it becomes the new head of the heap *k*th$_c$. The old head of the heap was r_x and is popped out of *K* and is inserted into the B_c list (lines 7–8). The update on the B_c list is triggered (line 9) which, in this case, does not affect the list. Similarly, when r_3 is inserted the same operations (lines 5–10) take place as with the insertion of r_2. Next, r_1 is inserted with the same effect, only this time the B_c list is altered during its update (line 9). r_2 is the new head of heap K_c and according to Definition 5 defines a new search space radius $d = dist(u_2, c) + 2 * radius_c$ (line 1 of Algorithm 3). The report r_x inside list B_c has $dist(u_x,\ c) > d$, thus it belongs to the tail of the list that is discarded in line 3 of Algorithm 3. When r_5 is inserted it ends up directly inside list B_c (line 12), since it is outside *c*, further away than *k*th$_c$ but closer than $dist(k\text{th}_c,\ c) + 2 * radius_c$ to the border of *c*. Reports r_0 and r_6 both end up directly inside list U_c (line 3), since they are covered by *c*, satisfying the condition in line 2.

After all reports are inserted into the k^+-heaps phase 1 of Algorithm 1 is completed and the search space is ready. For the second phase of Algorithm 1 the server scans a single k^+-heap for each user. The server can scan the k^+-heap of any *NCP* that covers a user *u* to get the *k* neighbors of *u*. In our Algorithm 1 the server

Table 2 Build-up phase of the k^+-heap of *NCP c* as user location reports are inserted

Arriving reports	Structure K$_c$	Structure B$_c$	Structure U$_c$	Line in algorithm 2
r_4	$\{r_4\}$	$\{\}$	$\{\}$	1,4,5
r_x	$\{r_x, r_4\}$	$\{\}$	$\{\}$	1,4,5
r_2	$\{r_4, r_2\}$	$\{r_x\}$	$\{\}$	1,4–11
r_3	$\{r_3, r_2\}$	$\{r_4, r_x\}$	$\{\}$	1,4–11
r_1	$\{r_2, r_1\}$	$\{r_3, r_4\}$	$\{\}$	1,4–11
r_5	$\{r_2, r_1\}$	$\{r_3, r_4, r_5\}$	$\{\}$	1,12,13
r_6	$\{r_2, r_1\}$	$\{r_3, r_4, r_5\}$	$\{r_6\}$	1–3
r_0	$\{r_2, r_1\}$	$\{r_3, r_4, r_5\}$	$\{r_6, r_0\}$	1–3

scans the *NCP* that actually services the user *ncp*(u) (lines 12). For users u_0 and u_6, the server Q scans $k_c^+ = u_2$, u_1, u_3, u_4, u_5, u_6 and finds nearest neighbors $\{u_2$, $u_1\}$ and $\{u_0$, $u_1\}$ for user u_0 and u_6 respectively.

5.6 Performance Analysis

In this section we analytically derive the performance of the *Proximity* framework in respect to computational complexity and scalability. We adopt worst-case analysis regarding user distribution and/or user movement pattern as it provides a bound for all input.

All computations in our framework happen on the server. *NCP*s do not participate in any processing; they just relay reports from the server to the mobile users and vice versa. We execute the query centrally on the server and assume that all the data can fit in main memory.

It is safe to say that in network setups, like the one we described in Sect. 3, the tendency is to maximize the users serviced n and minimize the number of network connectivity points $|C|$, as is the case for cellular network companies [44]. Therefore, we can assume that $|C| \ll n$. Furthermore, each *NCP* has a predefined communication capacity expressed in bits/sec [44]. Depending on the user traffic there is always a limit λ of the amount of users each *NCP* can serve [45]. λ is independent of n, since the capacity of the *NCP* and the user traffic profiles are independent of n. For simplicity we regard λ as a network parameter that is constant.

The following theorems show the correctness and the time complexity of *Proximity*.

Theorem 1 *For any user u covered by NCP cell c we only need to scan set S_c to find the actual k-nearest neighbors of u.*

Proof Let x denote some arbitrary user outside S_c, and q denote an arbitrary user in *NCP* cell c. We want to show that $dist(x, q) > dist(k\text{th}_q, q)$ always holds, where $k\text{th}_q$ is the kth nearest neighbor of q inside S_c.

Assume that $dist(x, q) \leq dist(k\text{th}_q, q)$. We will show that this leads to a contradiction. The worst case k-nearest neighbors of user q inside S_c are the users of K. In this case the kth nearest neighbor knn'_q will be $dist(knn'_q,\ q) < dist(k\text{th}_c,\ c) + 2 * radius_c$, where $k\text{th}_c$ is defined in Definition 3 (see Fig. 4). Putting the two inequalities together we would have $dist(x, q) \leq dist(knn'_q, q) \leq dist(k\text{th}_c, c) + 2 * radius_c$, which yields $dist(x, q) \leq dist(k\text{th}_c,\ c) + 2 * radius_c$. This is a contradiction as we assumed that user x is not inside S_c (Definition 5).

Lemma 1 *The build phase of Proximity has time complexity* $O(n * \log(k * \lambda))$.

Proof The build phase consists of $O(n * |C|)$ insertions into k^+-heaps. Insertion and deletion in heap K of a k^+-heap costs $O(\log k)$, since it is a conventional heap with constant size k. Insertion into the ordered list of size $|B|$ has worst case cost $O(\log|B|)$ using binary search. Similarly, inserting into and updating the *boundary* list costs $O(\log|B|)$. Thus, the worst case insertion cost for our novel k^+-heap is $O(\log k + \log|B|) = O(\log(k * |B|))$ and there are n insertions. Each *NCP* has a user limit λ and the boundary region B contains a finite number a of *NCP*s, thus $|B| = a * \lambda$. a is a finite number independent of n and $\alpha \ll |C| \ll n$. This makes $O(\log(k * |B|)) = O(\log(k * \lambda))$.

Lemma 2 *After all k^+-heaps are built, the scanning phase has time complexity* $O(n(k + \lambda))$.

Proof The size of a k^+-heap is $|S_c|$ and each user scans a k^+-heap (Theorem 1). Consequently we have $n * |S_c|$ comparisons. $|S_c| = k + |B_c| + |U_c|$ as defined by Definition 5. The size of U_c is bounded by the maximum number of users the *NCP* can serve $|U_c|_{\max} = \lambda$. $|B| = a * \lambda$ as described in proof of Theorem 1. Thus, $|S_c| = k + (a + 1) * \lambda$, which means that the time complexity of a single round of *Proximity* is $O(n(k + \lambda))$.

Theorem 2 *Each round of Proximity runs in* $O(n * \log(k * \lambda) + n(k + \lambda))$ *time.*

Proof Based on Theorem 1 and 2.

Using the *NCP*s for space partitioning, instead of a regular grid defined on the server, gives us the advantage of exploiting the user distribution adaptation that is inherent in the deployment of wireless or WiFi *NCP*s. It further frees us from setting a global parameter that would determine the size of the grid cell or a technique to adapt the grid size according to the user distribution, which would make our framework more complicated and possibly more time consuming.

6 Summary and Future Vision

We have motivated and defined the problem of continuously answering all k-nearest neighbor (CAkNN) queries and presented our algorithmic solution, called *Proximity*. Its efficiency is based on a division of the search space based on the network connectivity points and exploiting search space sharing among users of the same connectivity point. The *Proximity* framework has a better time complexity compared to solutions based on existing work. Our analysis verifies our arguments and shows that *Proximity* is very well suited for large scale scenarios. We have also provided insights of how the proposed CAkNN can serve as both a real-world benchmark and an improvement technique for existing computational intelligence methods.

Extensions and future plans for this work are also placed in parallelizing the *Proximity* algorithm, specializing it for cloud environments, and extending it to be a hierarchical *granular* algorithm. The first goal will allow for scalability necessary for the Big Data era. The latter goal will allow greater usability in disciplines of Neural Networks and Granular Computing [31].

References

1. Roussopoulos, N., Kelley, S., Vincent, F.: Nearest neighbor queries. In: Proceedings of the ACM SIGMOD international conference on management of data, ser. SIGMOD '95. New York, USA: ACM, pp. 71–79 (1995)
2. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (2006)
3. Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. Am. Stat. **46**(3), 175–185 (1992)
4. Hart, P.E.: The condensed nearest neighbor rule. IEEE Trans. Inf. Theory **18**, 515–516 (1968)
5. Shang, W., Huang, H., Zhu, H., Lin, Y., Wang, Z., Qu, Y.: An Improved kNN Algorithm—Fuzzy kNN. Computational Intell. Secur., Lect. Notes Comput. Sci. **3801**, 741–746 (2005)
6. Callahan, P.B.: Optimal parallel all-nearest-neighbors using the well-separated pair decomposition. In: Proceedings of the 1993 IEEE 34th annual foundations of computer science: IEEE Computer Society, pp. 332–340. Washington, DC (1993)
7. Clarkson, K.L.: Fast algorithms for the all nearest neighbors problem. Foundations of Computer Science, Annual IEEE Symposium on, vol. 83, pp. 226–232 (1983)
8. Gabow, H.N., Bentley, J.L., Tarjan, R.E.: Scaling and related techniques for geometry problems. In: Proceedings of the sixteenth annual ACM symposium on theory of computing, ser. STOC '84. New York ACM, pp. 135–143 (1984)
9. Lai, T.H., Sheng, M.-J.: Constructing euclidean minimum spanning trees and all nearest neighbors on reconfigurable meshes. IEEE Trans. Parallel Distrib. Syst. **7**(8), 806–817 (1996)
10. Wang, Y.-R., Horng, S.-J., Wu, C.-H.: Efficient algorithms for the all nearest neighbor and closest pair problems on the linear array with a reconfigurable pipelined bus system. IEEE Trans. Parallel Distrib. Syst. **16**, 193–206 (2005)
11. Chen, Y., Patel, J.: Efficient evaluation of all-nearest-neighbor queries, in Data Engineering. ICDE 2007. IEEE 23rd International Conference on, Apr. 2007, pp. 1056–1065 (2007)
12. Zhang, J., Mamoulis, N., Papadias, D., Tao, Y.: All-nearest-neighbors queries. In: International conference on spatial databases, scientific and statistical database management, vol. 0, p. 297 (2004)
13. Deb, K.: Multi-Objective optimization using evolutionary algorithms. Wiley, New York (2002)
14. Mao, J., Jain, K.: Artificial neural networks for feature extraction and multivariate data projection. IEEE Trans. Neural Netw. **6**(2), 296–317 (1995)

15. Hansen, P., Mladenovic, N.: Variable neighborhood search. In: Editors: Fred W Glover, Gary A Kochenberger.(eds.) Handbook of Metaheuristics, pp. 145–184. Kluwer, Netherlands (2003)
16. Zhang, Q., Li, H., MOEA/D.: A Multi-objective evolutionary algorithm based on decomposition. In: IEEE Transactions on evolutionary computation (2007)
17. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA II, IEEE TEC (2002)
18. Federal Communications Commission—Enhanced 911 website Jan 2014. [Online]. Available: http://www.fcc.gov/pshs/services/911-services/enhanced911/
19. Department of transportation: Intelligent transportation systems new generation 911 website Jan 2014. [Online]. Available. http://www.its.dot.gov/NG911/
20. Rayzit website (Jan 2014). [Online]. Available. http://www.rayzit.com
21. Waze website Jan 2014. [Online]. Available: `Waze.` http://www.waze.com/
22. Hoffer, J., Ramesh, V., Topi, H.: Modern database management (2013)
23. Smart metering entity website (Jan 2014). [Online]. Available. http://www.smi-ieso.ca/mdmr
24. Popular science: Inside google's quest to popularize self-driving cars article Jan 2014. [Online]. Available. http://www.popsci.com/cars/article/2013-09/google-self-driving-car
25. Lu, W., Shen, Y., Chen, S., Ooi, B.C.: Efficient processing of k nearest neighbor joins using mapreduce. Proc. VLDB Endow. **5**(10), 1016–1027 (2012)
26. Zhang, C., Li, F., Jestes, J.: Efficient parallel knn joins for large data in mapreduce. In: Proceedings of the 15th international conference on extending database technology, ser. EDBT '12. New York ACM, pp. 38–49 (2012)
27. Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. OSDI **2004**, 137–150 (2004)
28. Boehm, C., Krebs, F.: The k-nearest neighbour join: Turbo charging the kdd process. Knowl. Inf. Syst. **6**(6), 728–749 (2004)
29. Seiffert, U., Schleif, F.-M., Zühlke, D.: Recent trends in computational intelligence in life sciences In ESANN (2011)
30. Thomas, S., Jin, Y.: Reconstructing biological gene regulatory networks: where optimization meets big data, Evolutionary Intelligence, pp. 1–19 (2013)
31. Witold Pedrycz.: Granular computing: Analysis and design of intelligent systems. In CRC Press (2013)
32. Ranzato, Q.Le., Monga, R., Devin, M., Chen, K., Corrado, G., Dean, J., Ng, A.: Building high-level features using large scale unsupervised learning. In: International conference in machine learning (2012)
33. Hall, L.O., Chawla, N., Bowyer, K.W.: Decision tree learning on very large data sets. In: IEEE international conference on system, man and cybernetics (SMC), pp. 187–222 (1998)
34. Patil, D.V., Bichkar, R.S., A hybrid evolutionary approach to construct optimal decision trees with large data sets. In: IEEE international conference on industrial technology, pp. 429–433 (2006)
35. Lu, Y.-L., Fahn, C.-S.: Hierarchical artificial neural networks for recognizing high similar large data sets. In: International conference on machine learning and cybernetics, vol. 7, pp. 1930–1935 (2007)
36. Geolocation API website Jan 2014. [Online]. Available. http://code.google.com/apis/gears/api_geolocation.html
37. Vaidya, P.M.: An o(n log n) algorithm for the all-nearest-neighbors problem. Discrete, Computational Geom. **4**, 101–115 (1989)
38. Xia, C., Lu, H., Ooi, B.C., Hu, J., Gorder: an efficient method for knn join processing. In: Proceedings of the 13th international conference on Very large data bases—vol 30, ser. VLDB '04. VLDB Endowment, pp. 756–767 (2004)
39. Yao, B., Li, F., Kumar, P.: K nearest neighbor queries and knn-joins in large relational databases (almost) for free. In: Data engineering (ICDE), 2010 IEEE 26th international conference on, pp. 4–15 (2010)

40. Yu, C., Cui, B., Wang, S., Su, J.: Efficient index-based knn join processing for high-dimensional data. Inf. Softw. Technol. **49**(4), 332–344 (2007)
41. Yu, X., Q.K., Pu, Koudas, N.: Monitoring k-nearest neighbor queries over moving objects. In: Proceedings of the 21st international conference on data engineering ser. ICDE '05 IEEE computer society, pp. 631–642 Washington, DC (2005)
42. Mouratidis, K., Papadias, D., Hadjieleftheriou, M., Conceptual partitioning: an efficient method for continuous nearest neighbor monitoring. In: Proceedings of the ACM SIGMOD international conference on management of data, ser. SIGMOD '05. New York: ACM, pp. 634–645 (2005)
43. Chatzimilioudis, G., Zeinalipour-Yazti, D., Lee, W.-C., Dikaiakos, M. D.: Continuous all k-nearest neighbor querying in smartphone networks. In: 13th international conference on mobile data management (MDM'12) 2012
44. Rappaport, T.: Wireless communications: principles and practice, 2nd edn. Prentice Hall PTR, Upper Saddle River, NJ (2001)
45. Universal mobile telephone system world website Jan 2014. [Online]. Available. http://www.umtsworld.com/technology/capacity.htm

Information Mining for Big Information

Yuichi Goto

Abstract Knowledge discovery of database or data (KDD) is to acquire knowledge from data. Mining interesting patterns from data or information granules is the most important process of KDD. The post-process of the mining that is to acquire knowledge from the interesting patterns is also important. As obtained interesting patterns increases, it becomes hard for analysts to do the post-process of the mining because they have done the process empirically and manually. This chapter has investigated the post-process of the mining, and presented a support method and tools for the process. Information mining is a process to acquire knowledge from the interesting patterns discovered by mining from data or information granules. Consistent verification, information abstraction, hypothesis generation, hypothesis verification, and information deduction are activities of information mining. Current data mining methods and information granulation methods are suitable for information abstract, but not suitable for the other activities. The present author has shown that strong relevant logic-based reasoning is a systematic method for supporting information mining, and introduced a forward reasoning engine, a truth maintenance system, and epistemic programming can be used for support tools of the information mining with strong relevant logic-based reasoning.

Keywords Information mining · Strong relevant logics · Forward reasoning engine · Truth maintenance system · Epistemic programming

Y. Goto (✉)
Department of Information and Computer Sciences, Saitama University,
Saitama 338-8570, Japan
e-mail: gotoh@mail.saitama-u.ac.jp

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_2,

1 Introduction

The data-information-knowledge-wisdom hierarchy (DIKW hierarchy) is often used implicitly in definitions of data, information and knowledge [26]. *Data* is an elementary and recorded description of things, events, activities and transactions, lacks meaning or value, and is unorganized and unprocessed; *Information* is data processed to be meaningful, and valuable and appropriate for a specific purpose; *Knowledge* might be viewed as a mix of information and already obtained background knowledge (understanding, capability, experience, skills, values, and so on) [26]. Information is an intermediate between data and knowledge. Under the DIKW hierarchy, a process to mix obtained information with background knowledge is needed to acquire knowledge.

Knowledge discovery of database or data [15] (KDD) is to acquire knowledge from data. The word *data mining* is used for both KDD itself and a sub-process of a KDD process [19]. As a sub-process of a KDD process, data mining is a process to discover interesting patterns from massive amounts of data [15, 19]. A pattern is an expression in some language describing a subset of the data or a model applicable to the subset. An interesting pattern is a pattern that is interesting for some people in a specific domain. Interesting patterns are at least pieces of information because interesting patterns are processed data and they have meaning, from view point of the DIKW hierarchy. Interesting patterns may be adopted as pieces of knowledge directly, or they may be used as materials to acquire knowledge.

KDD with granular computing is to acquire knowledge from data via information granules. *Granular computing* is about representing, constructing, and processing information granules [27]. Informally, *information granules* can be treated as linked collections (clumps) of objects drawn together by the criteria of indistinguishability, similarity, proximity or functionality [29, 34, 35]. KDD with granular computing involves a process to discover interesting patterns from information granules [22]. Hereafter, we call the mining process *information granules mining* as like as data mining.

In the future, analysts will be able to discover the large number of interesting patterns from Big Data easily. *Big Data* is a term used to identify data sets that we cannot manage with current methodologies or software tools due to their large size and complexity [13]. Many researchers try to develop methodologies and software tools for data or information granules mining for Big Data. On the other hand, several companies and universities start to grow up data scientists [7], who are experts of KDD for Big Data. The data scientists with developed effective methodologies and tools will try to discover interesting patterns from Big Data. Moreover, the scale of Big Data and the number of kinds of data will increase [13, 24].

It is hard for analysts to acquire knowledge from the large number of interesting patterns without systematic and computer-assisted methods. In current KDD, analysts have done the process to acquire knowledge from interesting patterns empirically and manually. As the number of the interesting patterns increases, it becomes hard to do the process empirically and manually. Supporting the process is not focused on in Big Data mining [13, 19, 24].

In this chapter, the present author has investigated the post-process of the mining, and proposed a support method and tools for the process. *Information mining* is a process to acquire knowledge from the interesting patterns discovered by mining from data or information granules, and is a post-process of the mining processes. Current data mining methods and information granulation methods are not enough for information mining. Thus, the present author has shown that strong relevant logic-based reasoning is a systematic method for supporting information mining, and introduced a forward reasoning engine, a truth maintenance system, and epistemic programming can be used for support tools of the information mining with strong relevant logic-based reasoning.

The rest of the chapter is organized as follows: Sect. 2 explains information mining and the relationship between data mining and information mining; Sect. 3 introduces strong relevant logic-based reasoning as an information mining method; Sect. 4 shows a forward reasoning engine, a truth maintenance system, and epistemic programming language as support tools for information mining with strong relevant logic-based reasoning; Sect. 5 gives a summary and future works.

2 Information Mining in Knowledge Discovery from Data

A KDD process typically involves data cleaning, data integration, data selection, data transformation, data mining, pattern evaluation, and knowledge presentation [19]. *Data cleaning* is a process to remove noise and inconsistent data. *Data integration* is a process to combine multiple data sources. *Data selection* is a process to retrieve data relevant to the analysis task from integrated data set. *Data transformation* is a process to transform or consolidate data into forms appropriate for mining. *Data mining* is a process to extract interesting patterns from data set. *Pattern evaluation* is a process to identify the truly interesting patterns representing knowledge. *Knowledge presentation* is a process to transform the mined knowledge into understandable representation. In [15], there is another process that is “checking for and resolving potential conflicts with previously believed (or extracted) knowledge.” Here, we call the process as *consistent verification*. A KDD process is an iterative sequence of the above processes.

In KDD with granular computing, there are two other processes: information granulation and information granules mining. *Information granulation* is a process to transform or map plain data into information granules according to models based on fuzzy sets, rough sets, shadowed sets. *Information granules mining* is a process to discover interesting patterns from information granules. Figure 1 shows a control flow of processes in KDD with data mining and granular computing.

Analysts do other activities dealing with obtained interesting patterns and background knowledge after getting the interesting patterns in a KDD process. Those activities are information abstraction, hypothesis generation, hypothesis

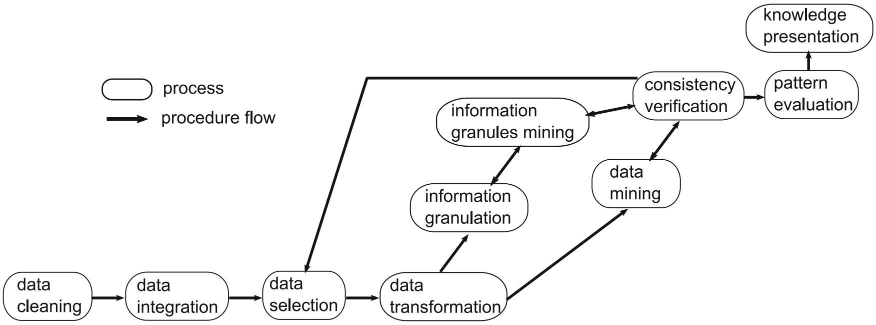


Fig. 1 Processes in knowledge discovery from data

verification, and information deduction. *Information abstraction* is to make interesting patterns more abstract. Information depends on a context and/or a person who interprets it. While a person sees a thing as valuable information, other person may see it as data with no particular significance. Knowledge is special information for a specific purpose, but meaning and value of knowledge are widely accepted by not only a certain person, but also many people. Therefore, the degree of universality of knowledge is higher than that of information. Moreover, as the degree of universality of information (knowledge) becomes higher, the value of information (knowledge) becomes higher. Abstraction is a way to increase the degree of universality. Thus, analysts make interesting patterns more abstract. *Hypothesis generation* is to create hypotheses from the interesting patterns and background knowledge. A KDD process can be regarded as a hypothetico-deductive process. Analysts, therefore, make hypotheses not only before starting KDD, but also during KDD. *Hypothesis verification* is to verify whether already proposed hypotheses are consistent with obtained interesting patterns. Of course, analysts should verify whether already proposed hypotheses are consistent with obtained data set. In addition, they should verify the consistency of proposed hypotheses in information level. *Information deduction* is to draw previously unknown or implicitly known information from obtained interesting patterns and background knowledge. Information deduction is a way to mix obtained interesting patterns with background knowledge.

Those activities are related to each other. Information deduction is done before doing consistent verification and hypothesis verification. If interesting patterns or hypotheses directly conflict with already obtained interesting patterns and background knowledge, analysts can find the conflicts by only checking the interesting patterns/hypotheses, already obtained interesting patterns, and background knowledge. If interesting patterns or hypotheses indirectly conflict with already obtained interesting patterns and background knowledge, analysts should deduce implicit things from the interesting patterns/hypotheses, already obtained interesting patterns, and background knowledge as premises, and check whether deduced things conflict with the premises. Moreover, to get more implicit information or hypotheses, information deduction is done after doing information

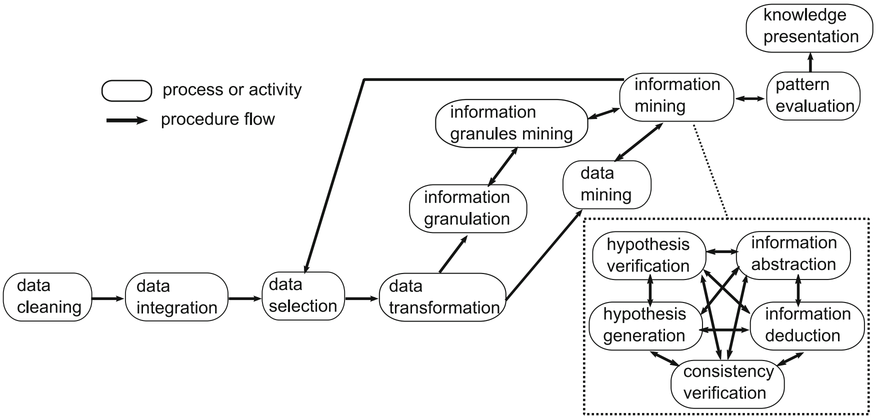


Fig. 2 Processes in knowledge discovery from data with information mining

abstraction or hypothesis generation. Those activities are done in an arbitrary order and iteratively.

Information mining is a process to acquire knowledge from obtained information patterns. The activities to deal with interesting patterns and background knowledge, i.e., consistent verification, information abstraction, hypothesis generation, hypothesis verification, and information deduction are activities of information mining because the activities are used in a process to acquire knowledge from obtained information patterns. The definition of information mining is similar to García-Martínez, et. al.'s definition, i.e., "Information Mining is the sub-discipline of information systems which supports business intelligence tools to transform information into knowledge" [16]. However, they used both data mining and information mining as a same meaning, and the processes of information mining they analyzed are processes from data integration to data mining in Fig. 1. In other words, the processes are until getting interesting patterns. Both definitions are similar, but purpose and scope are different. On the other hand, the other definition of information mining [32] is data mining dealing with unstructured data, i.e., data that are not stored in databases. In data mining for Big Data, dealing with unstructured data is as a matter of course [13, 24]. The definition and the present author's definition are different. Figure 2 shows a control flow of processes in KDD with information mining.

It will be hard for analysts to do information mining without systematic and computer-assisted methods as the number of interesting patterns increases. Analysts have done the information mining empirically and manually when they try to discover knowledge from data. However, as the number of the interesting patterns increases, it becomes difficult to do information mining empirically and manually. In the future, analysts will be able to discover the large number of interesting patterns from Big Data easily as we mentioned in Sect. 1. Supporting information mining is not focused on in Big Data mining [13, 19, 24].

Current data mining methods, information granules mining methods, and information granulation methods are useful for information abstraction, but not enough to do the other activities in information mining. The purpose of current data mining methods is to help pattern discovery [19, 33]. To discover patterns is a way of information abstraction. Information granulation is also a way of information abstraction. Thus, we can use data mining methods, information granules mining methods, and information granulation methods for information abstraction. However, they cannot be applied to the other activities of the information mining: consistency verification, hypothesis generation, hypothesis verification, and information deduction.

Requirements of hopeful information mining methods are as follows. (1) The method should be able to deal with all of information mining activities because analysts should do not only one activity but also several activities several times. (2) The method can be done automatically or semi-automatically. (3) Formalization method to describe interesting patterns and background knowledge in the method should have enough expressiveness because the outputs of the data mining methods and information granules mining methods are represented as various forms.

3 Strong Relevant Logic-Based Reasoning as an Information Mining Method

3.1 Deduction, Induction, and Abduction in Information Mining

Reasoning is helpful for consistency verification, information abstraction, hypothesis generation, hypothesis verification, and information deduction. Reasoning is the process of drawing new conclusions from given premises, which are already known facts or previously assumed hypotheses (Note that how to define the notion of new formally and satisfactorily is still a difficult open problem until now) [3, 4].

Reasoning can be classified into three forms, deductive reasoning, inductive reasoning, and abductive reasoning. Deductive reasoning (deduction) is the process of deducing or drawing conclusions from some general principles already known or assumed. Inductive reasoning (induction) is the process of inferring some general laws or principles from the observation of particular instances. Abductive reasoning (abduction) is the process whereby a surprising fact is made explicable by the application to it of a suitable proposition. The deductive reasoning guarantees that the conclusions deduced in the process are true if all premises are true, but inductive and abductive reasoning do not do that.

Deductive reasoning plays an important role for consistency verification and hypothesis verification. To do both verifications, it is necessary to check whether contradictions are drawn from discovered interesting patterns/given hypotheses and background knowledge. Analysts cannot know what contradictions occur so that it is difficult and ad hoc to check that proposition by using proving. By using deductive reasoning, analysts can check the proposition systematically. Doing information deduction is just deductive reasoning. Inductive reasoning is a way of information abstraction. Similarly, abductive reasoning is a way of hypothesis generation.

3.2 Formal Logic System and Formal Theory

To accept results of consistency verification, hypothesis generation, and information deduction with deductive reasoning as correct ones, deductive reasoning should be a logically valid reasoning. When doing deductive reasoning, deduced conclusions should be true if premises are true, and that the conclusions should be related to the premises. How can we ensure such feature of deductive reasoning? The answer is logics. A logically valid reasoning is a reasoning such that its arguments are justified based on some logical validity criterion provided by a logic system in order to obtain correct conclusions (Note that here the term correct does not necessarily mean true) [3, 4].

In general, a formal logic system L consists of a formal language, called the object language and denoted by $F(L)$, which is the set of all well-formed formulas of L, and a logical consequence relation, denoted by meta-linguistic symbol $\vdash_L$, such that $P \subseteq F(L)$ and $c \in F(L)$, $P \vdash_L c$ means that within the frame work of L, c is valid conclusion of premises P, i.e., c validly follows from P. For a formal logic system $(F(L), \vdash_L)$, a logical theorem t is a formula of L such that $\phi \vdash_L t$ where ϕ is empty set. Let $Th(L)$ denote the set of all logical theorems of L. $Th(L)$ is completely determined by the logical consequence relation $\vdash_L$. According to the representation of the logical consequence relation of a logic, the logic can be represented as a Hilbert style axiomatic system, a Gentzen natural deduction system, a Gentzen sequent calculus system, or other type of formal system. A formal logic system L is said to be *explosive* if and only if $\{A, \neg A\} \vdash_L B$ for any two different formulas A and B; L is said to be *paraconsistent* if and only if it is not explosive.

A formal theory with premises P based on L, called a L-theory with premises P and denoted by $T_L(P)$, is defined as $T_L(P) =_{\mathrm{df}} Th(L) \cup Th^{\mathrm{e}}_L(P)$, and $Th^{\mathrm{e}}_L(P) =_{\mathrm{df}} \{et \mid P \vdash_L et \text{ and } et \neq Th(L)\}$ where $Th(L)$ and $Th^{\mathrm{e}}_L(P)$ are called the logical part and the empirical part of the formal theory, respectively, and any element of $Th^{\mathrm{e}}_L(P)$ is called an empirical theorem of the formal theory. Figure 3 shows the relationship among $F(L)$, $T_L(P)$, $Th(L)$, and $Th^{\mathrm{e}}_L(P)$ of a formal logic system L. A formal theory $T_L(P)$ is said to be *directly inconsistent* if and only if there exists a formula A of L such that both $A \in P$ and $\neg A \in P$ hold. A formal theory $T_L(P)$ is said to be *indirectly inconsistent* if and only if it is not directly inconsistent but there exists a formula A of L such that both $A \in T_L(P)$ and $\neg A \in T_L(P)$; a formal theory $T_L(P)$ is said to be *consistent* if and only if it is neither directly inconsistent

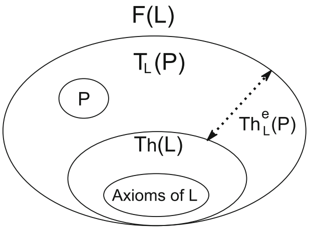


Fig. 3 *L*-theory with premises *P*

nor indirectly inconsistent. A formal theory $T_L(P)$ is said to be explosive if and only if $A \in T_L(P)$ for any $A \in F(L)$; $T_L(P)$ is said to be *paraconsistent* if and only if it is not explosive. An explosive formal theory is not useful at all. Therefore, any meaningful formal theory should be paraconsistent [3, 4]. Note that if a formal logic system L is explosive, then any directly or indirectly inconsistent L-theory $T_L(P)$ must be explosive.

3.3 Reasoning with Strong Relevant Logics

To do deductive reasoning based on a formal logic L for information mining is to obtain $Th^{\mathrm{e}}_L(P)$ where P is premises that represent already discovered patterns given by the mining activities. Today, there are so many different logic systems motivated by various philosophical considerations. As a result, a reasoning may be valid for one logical validity criterion but invalid for another. If logic systems underlying deductive reasoning are different, results of deductive reasoning may be different. In other words, although premises P is same, $Th^{\mathrm{e}}_L(P)$ and $Th^{\mathrm{e}}_{L'}(P)$ may be different where formal logic system L and L' are different. Thus, we have to choose a suitable logic system underlying deductive reasoning.

A logic system underlying deductive reasoning should ensure truth-preserving, relevant, ampliative, paracomplete, paraconsistent reasoning [3, 4]. Strong relevant logic and its family [3, 4] are hopeful candidates for logic systems underlying deductive reasoning in information mining process. Classical mathematical logic (CML for short) has been widely used for logic system underlying proof and reasoning. However, reasoning based on CML and its conservative extensions is truth-preserving, but not relevant, ampliative, and paraconsistent [1–4]. Thus, CML and its conservative extensions are not suitable for logic system underlying reasoning in information mining process. Relevant logics were constructed as logic systems that are more suitable for underlying reasoning rather than CML and its extensions [1, 2]. After that, as logic systems that are more suitable for underlying reasoning rather traditional relevant logics, strong relevant logics [3] were proposed. Then, family of strong relevant logics, e.g., temporal relevant logics, deontic relevant logics, spatial relevant logics, were also proposed [4]. Reasoning based on strong relevant logics and its family is truth-preserving, relevant, ampliative, paracomplete, and paraconsistent reasoning.

Meanwhile, logic-based formalization is suitable for representing already discovered interesting patterns in data mining or information granules mining. Basically, the discovered patterns denote the relationship among data objects, discovered classes, and the target data set [17]. Logic-based representation is suitable for describing such qualitative information. Especially, conditional relations, i.e., "if … then …," are useful for describing above relations. By using logic-based formalization, we can ignore the difference of approaches among data mining methods and information granules mining methods that are used to obtain patterns when we try to do information mining. Moreover, we can deal with background knowledge as well as discovered interesting patterns by using logic-based formalization.

Consequently, we can conclude that strong relevant logic-based reasoning is hopeful as a systematic method of the information mining. Strong relevant logic-based reasoning means (1) adopting strong relevant logics or its family as a logic system underlying reasoning, (2) formalizing target information into logical formulas based on the logic system, and (3) doing deductive, inductive, and abductive reasoning based on the logic system.

4 Supporting Tools for Information Mining with Strong Relevant Logic-Based Reasoning

4.1 Forward Reasoning Engine

As mentioned above in Sect. 3, to do deductive reasoning based on a formal logic L for information mining is to obtain a set of all empirical theorems $Th_L^e(P)$ where P is premises that represent already discovered patterns given by the mining activities; and empirical theorems are theorems in a target domain. Is there a support tool to obtain the $Th_L^e(P)$? That is a forward reasoning engine.

A forward reasoning engine is a computer program to automatically draw new conclusions by repeatedly applying inference rules, which are programmed in the reasoning engine or given by users to the reasoning engine as input, to given premises and obtained conclusions until some previously specified conditions are satisfied. The first forward reasoning engine is "Logic Theory Machine," developed by Newell, Shaw and Simon in 1957. As a well-known fact, the Logic Theory Machine was not successful due to the problem of computational complexity [8]. This (and the resolution method discovered by Robinson) led almost all researchers to adopt the more efficient approach of backward reasoning but not approach of forward reasoning [25]. However, from the viewpoint of logic validity of reasoning, the failure of Logic Theory Machine is caused by classical mathematical logic rather than forward reasoning [6] as mentioned above in Sect. 3. If we want to create, discover, or predict some new things rather that prove some things previously specified, then the only way is to ask forward reasoning. Forward reasoning engines support analysts to information mining activities by doing strong relevant logic-based reasoning automatically or semi-automatically.

FreeEnCal [6, 18] was proposed and developed as a forward reasoning engine with general-purpose, and is a hopeful candidate for a forward reasoning engine for information mining with strong relevant logic-based reasoning. It can interpret specifications written in the formal language such that any user can use the formal language to describe and represent formulas and inference rules for deductive, simple inductive, and simple abductive reasoning. It also can reason out all or a part of logical theorem schemata of a logic system, i.e., $Th(L)$ where L is a formal logic system as mentioned in Sect. 3, under the control conditions attached to the reasoning task specified by users, and all or a part of empirical theorems of a formal theory and facts, $Th_L^e(P)$ where P is premises, under the control conditions attached to the reasoning task specified by users. We can adopt FreeEnCal as a support tool for doing information mining with strong relevant logic-based reasoning.

4.2 Truth Maintenance System

An information mining process must be non-monotonic. In general, obtained interesting patterns and background knowledge may be incomplete and inconsistent. Moreover, inductive and abductive reasoning do not guarantee that the drawn conclusions are true if all premises are true. Thus, as information mining progresses, the amount of information may change because of solving contradictional information or reducing old or wrong information in the target cluster of information.

A truth maintenance system [10] (TMS for short), and also called a belief revision system or reason maintenance system was proposed to realize information systems to deal with such non-monotonic processes. A TMS works with an inference engine. The inference engine is a program to draw derived data from premises, assumptions, and other derived data, and it gives the derived data to the TMS (e.g., a forward reasoning engine is a kind of inference engines). *Premises* are used to define data that is always true. This data is not dependent on other data, and is not inferred from other facts. *Assumptions* are believed in the lack of evidence to the contrary, and are taken to be true until the contrary is proved. Premises, assumptions, and derived data managed in a TMS are called *beliefs*. If the TMS detects a contradiction in the current belief set stored in it, then the TMS eliminates it by revising the current belief set. When the TMS is revising the current belief set, it uses justifications of derived data for searching which assumptions are causes of the contradiction. *Justifications* describe the dependencies between data. The TMS gives all beliefs in the current belief set to the inference engine when the engine requires them. The main task of TMSs is to keep consistency of the current belief set stored in the TMSs.

The first TMS was proposed by Doyle [10]. After that, many TMSs were proposed. Stanojevic et al. [30] classified TMSs into three kinds: justification-based TMS (JTMS) [10], assumption-based TMS (ATMS) [9], and logic-based TMS (LTMS) [20, 21]. JTMSs and LTMSs can deal with only one context while ATMSs can deal with multi-context. A *context* is a set of all data that can be derived from an environment. An *environment* is a set of assumptions that uniquely describe a state. One context is uniquely determined by the corresponding environment. An environment is consistent if a contradiction cannot be inferred from the corresponding set of assumptions. To find which environment is inconsistent, ATMSs use labels. A *label* can be attached to each datum, describing which environment it will hold in. Label contains sets of environments in which the corresponding facts are valid. Using labels, we can immediately tell whether or not a datum holds under some assumptions. A *node* represents a data structure that usually contains an index (used to describe the node uniquely), a corresponding inference engine's datum, its justification (or justifications), and a label (in ATMSs and LTMSs). JTMSs and ATMSs do not require that premises, assumption, and derived data are represented as logical formulas while LTMSs require that to provide a facility of proof by refutation without inference engines. There are several extensions of ATMSs that focus on the uncertainty of assumptions [11, 12, 23, 28].

TMSs are a useful mechanism to support information mining with strong relevant logic-based reasoning. In the information mining, premises are background knowledge of a target domain and problem; assumptions are already discovered interesting patterns by mining activities, and drawn pieces of information by inductive reasoning or abductive reasoning; an inference engine is a forward reasoning engine that can deal with strong relevant logic-based reasoning. By using TMSs, it is possible to manage the consistency of the current set of pieces of information automatically or semi-automatically.

However, the above traditional TMSs are not suitable for cooperating with inference engines that do reasoning based on paraconsistent logics like strong relevant logics [17]. An operation to keep a consistency of current belief set is a primitive operation for traditional TMSs. Logic systems underlying traditional TMSs are classical mathematical logic (CML) or its conservative extensions. Reasoning based on those logics is inconsistent reasoning, i.e., the reasoning allows that everything follows from a contradiction. Thus, traditional TMSs should solve the inconsistency of the current belief set as soon as possible when contradictions are found in the belief set. Unlike reasoning based on CML and its conservative extensions, reasoning based on strong relevant logics and its family is paraconsistent. It allows that contradictions are in premises. An operation to keep a consistency of current belief set is not a primitive operation of TMSs for paraconsistent reasoning. There is a gap between traditional TMSs and TMSs for information mining with relevant logic-based reasoning. The present author has proposed the TMS for paraconsistent reasoning [17], and has been developing it.

4.3 Epistemic Programming

As the supporting tools, it will be necessary to prepare an environment to simulate epistemic processes that include a process of elimination, process of reduction to absurdity, and processes of deductive reasoning, inductive reasoning, and abductive reasoning by using a forward reasoning engine and a truth maintenance system. Information mining is not easy task as well as data mining and information granules mining because analysts do not know what information or knowledge there is in obtained interesting patterns and how they can extract such useful information or knowledge from the interesting patterns and background knowledge. The analysts may acquire the information or knowledge by trial and error. Therefore, supporting tools for reducing the cost of information mining will be demanded.

Epistemic programming [3] and its programming language [14, 31] can be used for constructing such simulation environment. A strong relevant logic model of epistemic processes in scientific discovery, and Epistemic programming was proposed as a novel program paradigm to program epistemic processes in scientific discovery.

Let $T_L(K)$ be an L-theory with premises K where $K \subseteq F(L)$ is a set of sentences to represent the explicitly known knowledge and/or current beliefs of an agent. An explicitly epistemic operation by the agent is any one of the following operations: for any $A \in T_L(K) - K$ where $T_L(K) \neq K$, an explicitly epistemic deduction of A from K, denoted by $K^{\mathrm{d}+A}$, is defined as $K^{\mathrm{d}+A} =_{\mathrm{df}} K \cup \{A\}$; for any $A \notin T_L(K)$ (note that we do not require $\neg \mathrm{A} \notin T_L(K)$), an explicitly epistemic expansion of K by A, denoted by $K^{\mathrm{e}+A}$, is defined as $K^{\mathrm{e}+A} =_{\mathrm{df}} K \cup \{A\}$, in particular, an explicitly epistemic simple-induction is an explicitly epistemic expansion $K^{\mathrm{e}+\forall x(A)}$ for $\exists x(A) \in K$ and an explicitly epistemic abduction is an explicitly epistemic expansion $K^{\mathrm{e}+A}$ for $C \in K$ and $A \Rightarrow \mathrm{C} \in K$ where $\Rightarrow$ denotes the notion of implication (entailment) in L; for any $A \in K$, an explicitly epistemic contraction of K by A, denoted by K^{-A}, is defined as $K^{-A} =_{\mathrm{df}} K - \{A\}$, in particular, an explicitly epistemic consistent-contraction is an explicitly epistemic contraction K^{-A} for $\neg A \in K$ or $K - \{\neg A\}$ for $A \in K$.

Let $T_L(K)$ be an L-theory with premises K where $K \subseteq F(L)$ is a set of sentences to represent the explicitly known knowledge and/or current beliefs of an agent. An implicitly epistemic operation by a forward reasoning engine (e.g., FreeEnCal) is any one of the following operations: or any K, an implicitly epistemic deduction of K, denoted by K^{d}, is defined as $K^{\mathrm{d}} =_{\mathrm{df}} T_L(K)$; for any $A \notin T_L(K)$ (note that we do not require $\neg A \notin T_L(K)$), an implicitly epistemic expansion of K by N to deduce A, denoted by $T_L(K \cup N)^{\mathrm{e}+A}$, is defined as $T_L(K \cup N)^{\mathrm{e}+A} =_{\mathrm{df}} T_L(K \cup N)$ where $N \subseteq F(L)$ such that $A \notin T_L(K)$ but $A \in T_L(K \cup N)$; in particular, an implicitly epistemic simple-induction is an implicitly epistemic expansion $T_L(K \cup N)^{\mathrm{e}+\forall x(A)}$ for $\exists x(A) \in T_L(K)$ and an implicitly epistemic abduction is an implicitly epistemic expansion $T_L(K \cup N)^{\mathrm{e}+\forall x(A)}$ for $C \in T_L(K)$ and $A \Rightarrow \mathrm{C} \in T_L(K)$ where $\Rightarrow$ denotes the notion of implication (entailment) in L; for any $A \in T_L(K)$, an implicitly epistemic contraction of K by N to delete A, denoted by $T_L(K - N)^{-A}$, is defined as $T_L(K - N)^{-A} =_{\mathrm{df}} T_L(K - N)$ where $N \subseteq K$ such that $A \notin T_L(K - N)$, in particular, an implicitly epistemic consistent-contraction is an implicitly epistemic contraction $T_L(K - N)^{-A}$ for $\neg A \in T_L(K)$ or $T_L(K - N)^{-A}$ for $A \in T_L(K)$.

Thus, an epistemic process of deductive-inductive-abductive belief revision can be defined as a sequence K_0, o_1, K_1, o_2, K_2, …, K_{n+1}, o_n, K_n where $K_i \subseteq F(L)$ $(0 \leq i \leq n)$, called an epistemic state of the epistemic process, is a set of sentences to represent known knowledge and current beliefs of an agent, and o_{i+1} $(0 \leq i \leq n)$, is any of explicitly or implicitly epistemic operations, and $K_{\ i+1}$ is the result of applying o_{i+1} to K_i. In particular, K_0 is called the primary epistemic state of the epistemic process, and K_n is called the terminal epistemic state of the epistemic process, respectively.

Note that the above definitions of epistemic operations and epistemic processes are general but not dependent on any special logic system. In [3], strong relevant logics are used for logic systems underlying the strong relevant logic model. Then, temporal relevant logics [4] are used for the model [5]. We can choose a suitable logic system in a family of strong relevant logics as a logic system underlying the model according to a target problem.

An epistemic program is a sequence of instructions such that for a primary epistemic state given as the initial input, an execution of the instructions produces an epistemic process where every epistemic operation corresponds to an instruction whose execution results in an epistemic state, in particular, the terminal epistemic state is also called the result of the execution of the program. We say that an epistemic program *replays* a scientific discovery if the execution of the program produces the same result as that discovered by the original discoverer in history when the program as input takes the same initial conditions as the original discoverer did. We say that an epistemic program *creates* or *makes* a scientific discovery if the execution of the program produces a result that is new, important, and interesting to the scientists working on the particular domain under investigation. An information mining process can be regarded as an epistemic process of scientific discovery process. By using Epistemic programming and its programming language, analysts can construct an environment to simulate or help to do own information mining processes.

Study of Epistemic programming is ongoing work. EPLAS was proposed as the first epistemic programming language [31], and its implementation was also proposed and developed [14]. A forward reasoning engine and a truth maintenance system are parts of the implementation of EPLAS. However, current EPLAS and its implementation provide poor representation power. For example, the current EPLAS and its implementation do not provide scientists with a high-level and general-purpose mechanism to deal with belief revision [17]. As a result, users of EPLAS have to program their belief revision processes by primary epistemic operations. That is not an easy task. To use EPLAS and its implementation for constructing the support environment for information mining, it is necessary to enrich representation power of them.

5 Summary

The chapter has investigated information mining as a new challenging issue in knowledge discovery from data (KDD) for Big Data. Information mining is a process to acquire knowledge from interesting patterns discovered by the mining from data or information granules, and is a post-process of the mining processes in a KDD process. Consistent verification, information abstraction, hypothesis generation, hypothesis verification, and information deduction are activities of information mining. In current KDD, analysts have done the information mining empirically and manually. However, it will be hard to do the information mining without systematic and computer-assisted methods as the number of interesting patterns increases. Current data mining methods and information granulation methods are suitable for information abstract, but not suitable for the other activities of information mining. The chapter has shown that strong relevant logic-based reasoning is a systematic method for supporting information mining, and introduced a forward reasoning engine, a truth maintenance system, and epistemic programming can be used for support tools of the information mining with strong relevant logic-based reasoning.

This is an ongoing work. Current epistemic programming language and its implementation are not enough to support for information mining. To improve and implement them are future works. Information granulation is a good way of information abstraction. Thus, to integrate information granulation and strong relevant logic-based reasoning is also a future work.

References

1. Anderson, A.R., Belnap Jr, N.D.: Entailment: the Logic of Relevance and Necessity, vol. 1. Princeton University Press, Princeton (1975)
2. Anderson, A.R., Belnap Jr, N.D., Dunn, J.M.: Entailment: the Logic of Relevance and Necessity, vol. 2. Princeton University Press, Princeton (1992)
3. Cheng, J.: A strong relevant logic model of epistemic processes in scientific discovery. In: Kawaguchi, E., Kangassalo, et al. (eds.) Information Modeling and Knowledge Bases XI. Frontiers in Artificial Intelligence and Applications, vol. 61, pp. 136–159. IOS Press, Amsterdam (2000)
4. Cheng, J.: Strong relevant logic as the universal basis of various applied logics for knowledge representation and reasoning. In: Kiyoki, Y., et al. (eds.) Information Modeling and Knowledge Bases XVII. Frontiers in Artificial Intelligence and Applications, vol. 136, pp. 310–320. IOS Press, Amsterdam (2006)
5. Cheng, J.: A temporal relevant logic approach to modeling and reasoning about epistemic processes. In: 2009 fifth international conference on semantics, knowledge and grids (SKG 2009), pp. 19–25. IEEE Computer Society, Beijing (2009)
6. Cheng, J., Nara, S., Goto, Y.: FreeEnCal: a forward reasoning engine with general-purpose. In: Apolloni, B., Howlett, R.J., Jain, L.C. (eds.) Knowledge-Based Intelligent Information and Engineering Systems, 11th International Conference, KES 2007, XVII Italian Workshop on Neural Networks, Vietri sul Mare, Italy, 12–14, Sept 2007. Proceedings, Part II. Lecture

Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), vol. 4693, pp. 444–452. Springer, Berlin (2007)

7. Davenport, T.H., Patil, D.J.: Data scientist: the sexiest job of the 21st century. Harvard Bus. Rev. **90**(10), 70–77 (2012)
8. Davis, M.: The early history of automated deduction. In: Robinson, A., Voronkov, A. (eds.) Handbook of Automated Reasoning, pp. 5–15. Elsevier and MIT Press, Amsterdam/Cambridge (2001)
9. de Kleer, J.: An assumption-based TMS. Artif. Intell. **28**(2), 127–162 (1986)
10. Doyle, J.: A truth maintenance system. Artif. Intell. **12**(3), 231–272 (1979)
11. Dubois, D., Berre, D.L., Prade, H., Sabbadin, R.: Using possibilistic logic for modeling qualitative decision: ATMS-based algorithms. Fundamenta Informaticae **37**(1–2), 1–30 (1999)
12. Dubois, D., Lang, J., Prade, H.: Handling uncertain knowledge in an ATMS using possibilistic logic. In: Ras, Z., Emrich, M. (eds.) Methodologies for Intelligent Systems, 5: Proceedings of the 5th International Symposium on Methodologies for Intelligent Systems Held 25–27 Oct 1990, pp. 252–259. Elsevier, Amsterdam (1990)
13. Fan, W., Bifet, A.: Mining big data: current status, and forecast to the future. ACM SIGKDD Explor. Newslett. **14**(2), 1–5 (2012)
14. Fang, W., Takahashi, I., Goto, Y., Cheng, J.: Practical implementation of EPLAS: an epistemic programming language for all scientists. In: 2011 international conference on machine learning and cybernetics (ICMLC 2011), pp. 608–616. IEEE, Guilin (2011)
15. Fayyad, U., Piatetsky-shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. AI Mag. **17**(3), 37–54 (1996)
16. García-Martínez, R., Britos, P., Rodríguez, D.: Information mining processes based on intelligent systems. In: Ali, M., Bosse, T., Hindriks, K., Hoogendoorn, M., Jonker, C., Treur, J. (eds.) Recent Trends in Applied Artificial Intelligence, 26th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2013, Amsterdam, The Netherlands, 17–21 June 2013. Proceedings. Lecture Notes in Computer Science, vol. 7906, pp. 402–410. Springer, Berlin (2013)
17. Goto, Y., Cheng, J.: A truth maintenance system for epistemic programming environment. In: 2012 eighth international conference on semantics, knowledge and grids (SKG 2012), pp. 1–8. IEEE Computer Society, Beijing (2012)
18. Goto, Y., Koh, T., Cheng, J.: A general forward reasoning algorithm for various logic systems with different formalizations. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) Knowledge-Based Intelligent Information and Engineering Systems, 12th International Conference, KES 2008, Zagreb, Croatia. 3–5 Sept 2008. Proceedings, Part II. Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), vol. 5178, pp. 526–535. Springer, Berlin (2008)
19. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques, 3rd edn. Morgan Kaufmann Publishers, Burlington, MA (2011)
20. McAllester, D.A.: An Outlook on Truth Maintenance. AI Memos 551 (1980)
21. McDermott, D.: A general framework for reason maintenance. Artif. Intell. **50**(3), 289–329 (1991)
22. Mitra, S., Pal, S.K., Mitra, P.: Data mining in soft computing framework: a survey. IEEE Trans. Neural Netw. **13**(1), 3–14 (2002)
23. Monai, F.F., Chehire, T.: Possibilistic assumption based truth maintenance system, validation in a data fusion application. In: The eighth international conference on uncertainty in artificial intelligence (UAI92), pp. 83–91. Morgan Kaufmann Publishers, Stanford, CA (1992)
24. O'Leary, D.E.: Artificial intelligence and big data. IEEE Intell. Syst. **28**(2), 96–99 (2013)
25. Robinson, A., Voronkov, A. (eds.): Handbook of Automated Reasoning, vol. 1–2. Elsevier and MIT Press, Amsterdam/Cambridge (2001)
26. Rowley, J.: The wisdom hierarchy: representations of the DIKW hierarchy. J. Inf. Sci. **33**(2), 163–180 (2007)

27. Pedrycz, W.: Granular Computing: Analysis and Design of Intelligent Systems. CRC Press/ Taylor & Francis, Boca Roton (2013)
28. Shen, Q., Zhao, R.: A credibilistic approach to assumption-based truth maintenance. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **41**(1), 85–96 (2011)
29. Skowron, A., Stepaniuk, J.: Information granules: towards foundations of granular computing. Int. J. Intell. Syst. **16**(1), 57–85 (2001)
30. Stanojevic, M., Vranes, S., Velasevicngine, D.: Using truth maintenance systems: a tutorial. IEEE Intell. Syst. **9**(6), 46–56 (1994)
31. Takahashi, I., Nara, S., Goto, Y., Cheng, J.: EPLAS: an epistemic programming language for all scientists. In: Shi, Y. (ed.) Computational Science—ICCS 2007: 7th International Conference, Beijing, China. 27–30 May 2007. Proceedings, Part I. Lecture Notes in Computer Science, vol. 4487, pp. 406–413. Springer, Berlin (2007)
32. Tkach, D.S.: Information mining with the IBM intelligent miner family. In: An IBM Software Solutions White Paper. pp. 1–29. IBM (1998)
33. Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., et al.: Top 10 algorithms in data mining. Knowl. Inf. Syst. **14**(1), 1–37 (2008)
34. Zadeh, L.: Fuzzy logic = computing with words. IEEE Trans. Fuzzy Syst. **4**(2), 103–111 (1996)
35. Zadeh, L.: Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets Syst. **90**(2), 111–127 (1997)

Information Granules Problem: An Efficient Solution of Real-Time Fuzzy Regression Analysis

Azizul Azhar Ramli, Junzo Watada and Witold Pedrycz

Abstract Currently, Big Data is one of the common scenario which cannot be avoided. The presence of the voluminous amount of unstructured and semi-structured data would take too much time and cost too much money to load into a relational database for analysis purpose. Beside that, regression models are well known and widely used as one of the important categories of models in system modeling. This chapter shows an extended version of fuzzy regression concept in order to handle real-time data analysis of information granules. An ultimate objective of this study is to develop a hybrid of a genetically-guided clustering algorithm called genetic algorithm-based Fuzzy C-Means (GAFCM) and a convex hull-based regression approach, which is regarded as a potential solution to the formation of information granules. It is shown that a setting of Granular Computing with the proposed approach, helps to reduce the computing time, especially in case of real-time data analysis, as well as an overall computational complexity. Additionally, the proposed approach shows an efficient real-time processing of information granules regression analysis based on the convex hull approach in which a Beneath-Beyond algorithm is employed to design sub-convex hulls as well as a main convex hull structure. In the proposed design setting, it was

A. A. Ramli
Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia
e-mail: azizulr@uthm.edu.my

J. Watada (✉)
Graduate School of Information, Production and Systems, Waseda University,
2-7, Hibikino, Wakamatsu-ku, Kitakyushu-shi, Fukuoka-ken 808-0135, Japan
e-mail: junzow@osb.att.ne.jp; junzo.watada@gmail.com

W. Pedrycz
Department of Electrical and Computer Engineering, University of Alberta, Edmonton,
AB T6G 2V4, Canada
e-mail: wpedrycz@ualberta.ca

W. Pedrycz
Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_3,

emphasized a pivotal role of the convex hull approach or more specifically the Beneath-Beyond algorithm, which becomes crucial in alleviating limitations of linear programming manifesting in system modeling.

Keywords Granular computing · Fuzzy regression analysis · Information granules · Fuzzy C-means · Convex hulls · Convex hull · Beneath-beyond algorithm

1 Introduction

Nowadays, a significant growth of interest in Granular Computing (GrC) is regarded as a promising vehicle supporting the design, analysis and processing of information granules [1]. With regard of all processing faculties, information granules are collections of entities (elements), usually originating at the numeric level, which are arranged together due to their similarity, functional adjacency and in distinguishability or alike [1]. Given the similarity function to quantify the closeness between the samples, these data are clustered into certain granules, categories or classes [2]. The process of forming information granules is referred as information granulation.

GrC has begun to play important roles in bioinformatics, pattern recognition, security, high-performance computing and others in terms of efficiency, effectiveness, robustness as well as a structural representation of uncertainty [2]. Therefore, the need for sophisticated Intelligent Data Analysis (IDA) tools becomes highly justifiable when dealing with this type of information.

The above statement supported with the amount of data generated by social media, transactions, public and corporate entities, whose amount is scaled faster than computer resources allow (Big Data scenario). Add to that challenge, the volume of data is generated by Internet of Thing (IoT) such like smartphones, tablets, PCs or smart glasses; it becomes clear that traditional solutions of data storage and processing could hardly be applied to ingest, validate and analyze these volumes of data [3].

Accordingly, the developed method discussed here exhibits sound performance as far as computing time and an overall computation complexity are concerned. Fuzzy C-Means (FCM) clustering algorithm, introduced by Dunn in 1973 [4] and generalized by Bezdek in 1981, becomes one of the commonly used techniques of GrC when it comes to the formation of information granules [5, 6]. There has been a great deal of improvements and extensions of this clustering technique. One can refer here to the genetically-guided clustering algorithm called Genetic Algorithm-FCM (GA-FCM) and proposed by Hall et al. [4]. It has been shown that the GA-FCM algorithm can successfully alleviate the difficulties of choosing a suitable initialization of the FCM method. On the other hand, Ramli et al. proposed a real-time fuzzy regression model incorporating a convex hull method, specifically a Beneath-Beyond algorithm [7]. They have deployed a convex hull approach useful in the realization of data analysis in a dynamic data environment.

Associated with these two highlighted models (fuzzy regression and fuzzy clustering), the main objective of this study is to propose an enhancement of the fuzzy regression analysis for the purpose of analysis of information granules. From the IDA perspective, this research intends to augment the model that Bezdek given originally proposed by including the Ramli et al.'s approach. It will be shown that such a hybrid combination is capable of supporting real-time granular based fuzzy regression analysis.

In general, the proposed approach helps perform real time fuzzy regression analysis realized in presence of information granules. The proposed approach comprises four main phases. First, the use of GA-FCM clustering algorithm granulates the entire data set into a limited number of chunks-information granules. The second phase consists of constructing sub-convex hull polygons for the already formed information granules. Therefore, the number of constructed convex hulls should be similar to the number of identified information granules. Next, main convex hull is constructed by considering all sub convex hulls. Moreover, the main convex hull will utilize the outside vertices which were selected from the constructed sub-convex hulls. Finally, in the last phase, the selected vertices of the main constructed convex hull, which covers all sub-convex hull (or identified in-formation granules), are used to build a fuzzy regressions model. To illustrate the efficiency and effectiveness of the proposed method, a numeric example is presented.

This chapter is structured as follows. Section 2 serves as a concise and focused review of the fundamental principles of real-time data analysis, GrC as well as GA-FCM. Furthermore, this section also highlighted a review on convex hull approach; affine, supporting hyperplane as well as Beneath-Beyond algorithm. Additionally, some essentials of fuzzy linear regression augmented by the convex hull approach have been discussed. Section 3 discusses a processing flow of the proposed approach yielding real time granular based fuzzy regression models. Section 4 is devoted to a numerical experiment. Finally, Sect. 5 presents some concluding remarks.

2 Some Related Studies

Through this section, several fundamental issues to be used throughout the study are investigated.

2.1 Recall of Real-Time Data Analysis Processing

Essentially, real-time data analysis refers to studies where data revisions (updates, successive data accumulation) or data release timing is important to a significant degree. The most important properties of real-time data analysis are dynamic analysis and reporting, based on data entered into a system in a short interval before the actual time of the usage of the results [8].

An important notion in real-time systems is event, that is, any occurrence that results in a change in the sequential flow of program execution. Related to this situation, the time between the presentation of a set of inputs and the appearance of all the associated outputs (results) is called the response time [9, 10]. In addition, the shortest response time is an important design requirement.

2.2 Brief Review on Granular Information

Granular Computing (GrC) is a general computing paradigm that effectively deals with designing and processing information granules. The underlying formalism relies on a way in which information granules are represented; here it may consider set theory, fuzzy sets, rough sets, to name a few of the available alternatives [1]. In addition, GrC focuses on a paradigm of representing and processing information in a multiple level architecture. Furthermore, GrC can be viewed as a structured combination of algorithmic and non-algorithmic aspects of information processing [5].

Generally, GrC is a twofold process and includes granulation and computation, where the former transforms the problem domain to the one with granules, whereas the latter processes these granules to solve the problem [11]. Granulation of information is an intuitively appealing concept and appears almost everywhere under different names, such as chucking, clustering, partitioning, division or decomposition [12]. Moreover, the process of granulation and the nature of information granules imply certain formalism that seems to be the most suited to capture the problem at hand. Therefore, to deal with the high computational cost which might be caused by a huge size of information granule patterns, it was noted that FCM algorithm which is one of commonly selected approaches to data clustering implementation procedure.

In general, the problem of clustering is that of finding a partition that captures the similarity among data objects by grouping them accordingly in the partition (or cluster). Data objects within a group or cluster should be similar; data objects coming from different groups should be dissimilar. In this context, FCM arises as a way of formation of information granules represented by fuzzy sets [5]. Clustering approach as well as the FCM clustering algorithm have been discussed in this section.

Clustering is a process of grouping a data set in a way that the similarity between data within same cluster is maximized while the similarity of data between different clusters is minimized [13]. It classifies a set of observations into two or more mutually exclusive unknown groups based on combinations of many variables. Its aim is to construct groups in such a way that the profiles of objects in the same groups are relatively homogeneous whereas the profiles of objects in different groups are relatively heterogeneous [13].

FCM is a method of clustering which allows any data to belong to two or more clusters with some degrees of membership. Initially, consider a data set composed of n vectors $X = \{x_1, x_2, \ldots, x_n\}$ to be clustered into c clusters or groups. Each of

$x_k \in \Re^K, k = 1, 2, \ldots, n$ is a feature vector consisting of K real-valued measurements describing the features of the objects. A fuzzy c-partition of the given data set is the fuzzy partition matric $U = [\mu_{ik}], i = 1, 2, \ldots, c$ and $k = 1, 2, \ldots, n$ such that

$$\begin{aligned} &0 \le \mu_{ik} \le 1, && for\ 1 \le i \le c, 1 \le k \le n \\ &0 \le \sum_{k=1}^{n} \mu_{ik} \le n, && for\ 1 \le i \le c \\ &\sum_{i=1}^{c} \mu_{ik} = 1, && for\ 1 \le k \le n \end{aligned} \tag{1}$$

where μ_{ik} is the membership of feature vector x_k to cluster c_i. Furthermore, fuzzy cluster of the objects can be represented by a membership matrix called fuzzy partition. The set of all $c \times n$ non-degenerate constrained fuzzy partition matrices denoted by M_{fcn} which is defined as

$$M_{fcn} = \left\{ U \in \Re^{c\times n} \middle| \sum_{i=1}^{c} = 1, 0 < \sum_{k=1}^{n} U_{ik} < n, U_{ik} \in [0,1]; 1 \le i \le c; 1 \le k \le n \right\}. \tag{2}$$

Moreover, the FCM algorithm minimizes the following objective function

$$J_m(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} (\mu_{ik})^m D_{ik}^2(\boldsymbol{v}_i, \boldsymbol{x}_k) \tag{3}$$

where $U \in M_{fcn}$ is a fuzzy partition matrix, $V = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c)$ is a collection of cluster centers (*prototypes*). $\boldsymbol{v}_i \in \Re^K \forall i$ and $D_{ik}(\boldsymbol{v}_i, \boldsymbol{x}_k)$ is a distance between $\boldsymbol{x}_k$ and the ith prototype while m is a fuzzification coefficient, $m > 1$.

The FCM optimizes (3) by iteratively updating the prototypes and the partition matrix. More specifically, some values of c, m and ε (termination condition—a small positive constant) have been chosen, then generate a random fuzzy partition matrix U^0 and set an iteration index to zero, $t = 0$. An iterative process is organized as follows. Given the membership value $\mu_{ik}^{(t)}$, the cluster centers $\boldsymbol{v}_i^{(t)} (i = 1, \ldots, c)$ are calculated by

$$\boldsymbol{v}_i^{(t)} = \frac{\sum_{k=1}^{n} \left(\mu_{ik}^{(t)}\right)^m \boldsymbol{x}_k}{\sum_{k=1}^{n} \left(\mu_{ik}^{(t)}\right)^m} \tag{4}$$

Given the new cluster centers $\boldsymbol{v}_i^{(t)}$ the membership values of the partition matrix $\mu_{ik}^{(t)}$ are updated as

$$\mu_{ik}^{t+1} = \left[\sum_{j=1}^{c} \left(\frac{\left\| \boldsymbol{x}_k - \boldsymbol{v}_i^{(t)} \right\|}{\left\| \boldsymbol{x}_k - \boldsymbol{v}_j^{(t)} \right\|} \right)^{\frac{2}{m-1}} \right]^{-1} \tag{5}$$

This process terminates when $|U^{(t+1)} - U^{(t)}| \leq \varepsilon$, or some predefined number of iterations has been reached [14]. In the following sub section, an enhancement of the FCM algorithm called GA-FCM is investigated.

2.3 Genetically-Guided Clustering Algorithm

There are several studies employed genetic algorithm based clustering technique in order to solve various types of problems [15–18]. More specifically, GA technique to determine the prototypes of the clusters located in the Euclidean space $\Re^K$ has been exploited. At each generation, a new set of prototypes is created through the process of selecting individuals according to their level of fitness. In the sequel they are affected by running genetic operators [16, 18]. This process leads to the evolution of population of individuals that become more suitable given the corresponding values of the fitness function.

There are a number of research studies that have been completed which utilizing the advantages of GA-enhanced FCM. Genetically guided clustering algorithm proposed by Hall et al. was focused here. Based on [4], in any generation, element i of the population is V_i, a $c \times s$ matrix of cluster centers (*prototypes*). The initial population of size P is constructed by a random assignment of real numbers to each of the s features of the c centers of the clusters. The initial values are constrained to be in the range (*determined from the data set*) of the feature to which they are assigned.

In addition, as V's will be used within the GA, it is necessary to reformulate the objective function for FCM for optimization purposes. Expression (3) can be expressed in terms of distances from the prototypes (as done in the FCM method). Specifically, for $m > 1$ as long as $D_{jk}(\boldsymbol{v}_j, \boldsymbol{x}_k) > 0 \, \forall j, k$, it have

$$\mu_{ik} = 1 \Big/ \sum\nolimits_{j=1}^{c} \left(\frac{D_{ik}(\boldsymbol{v}_i, \boldsymbol{x}_k)}{D_{jk}(\boldsymbol{v}_j, \boldsymbol{x}_k)} \right)^{\frac{2}{m-1}} \quad \textit{for} \quad 1 \leq i \leq c; 1 \leq k \leq n. \tag{6}$$

Now, Eq. (6) was substituted into Eq. (2). This gives rise to the FCM functional reformulated as follows

$$R_m(V) = \sum_{k=1}^{n} \left(\sum_{i=1}^{c} D_{ik}^{1/(1-m)} \right)^{1-m}. \tag{7}$$

which is concentrated on optimizing R_m with a genetically-guided algorithm (GGA) technique [4]. Additionally, Hathaway and Bezdek (1995) highlighted that have shown that local (V) minimizers of R_m and also (U) at expression (3) will produces local minimizers of J_m and, on the other hands, the V part of local minimizers of J_m acquiesce local minimizers of R_m [4].

Furthermore, there are a number of genetic operators, which relate to the GA-based clustering algorithm including *Selection* which consist of selecting parents for reproduction, performing *Crossover* with the parents and applying *Mutation* to the bits of the children [4]. Binary gray code representations where any two adjacent numbers are one bit different has been selected on this genetically-guided algorithm (GGA) approach and this encoding able to yields faster convergence and improve performance over a straightforward binary encoding [4].

The complete process of the GGA [4] is outlined as follows.

GGA1: Choose m, c, and D_{ik}.

GGA2: Randomly initialize P sets of c cluster centers. Confine the initial values within the space defined by the data to be clustered.

GGA3: Calculate R_m by using (7) for each population member and apply modified objective function $R'_m(V) = R_m(V) + b \times R_m(V)$ where $b \in [0, c]$ is the number of empty clusters.

GGA4: Convert population members to binary equivalents (using the Gray code).

GGA5: For $i = 1$ to number of generations, Do

(i) Used k-fold tournament selection (default $k = 1$) to select P/2 parent pairs for reproduction.

(ii) Complete a two-point crossover and bitwise mutation for each feature of the parent pairs.

(iii) Calculate R_m by using (7) for each population member and apply modified objective function $R'_m(V) = R_m(V) + b \times R_m(V)$ where b ∊ [0, c] is the number of empty clusters.

(iv) Create a new generation of size P, which is selected from the two best members of the previous generation and the best children that are generated by using crossover and mutation.

GGA6: Provide the cluster centers to the terminal population with the smallest R'_m value and report R'_m.

2.4 A Brief Review of a Convex Hull Approach

The convex hull is the fundamental construct of mathematics and computational geometry. It is useful as a building block for a plethora of applications including collision detection in video games, visual pattern matching, mapping and path determination [19]. In what follows, a detailed description of this approach was presented.

2.4.1 Affine, Convex Hull Definition and Supporting Hyperplane

The affine hull of set S in Euclidean space $\Re^K$ is the smallest affine set contained in S, or equivalently the intersection of all the affine sets containing S. Here, an affine set is defined as the translation of a vector subspace. The affine hull $aff(S)$ of S is the set of all the affine combinations of elements of S, namely

$$aff(S) = \left\{ \sum_{j=1}^{K} \alpha_j x_j \middle| x_j \in S, \alpha_j = \Re, \alpha_j \geq 0, \sum_{j=1}^{K} \alpha_j = 1 \right\}. \tag{8}$$

The convex hull of set S of points $hull(S)$ is defined to be a minimal convex set containing S. A point $P \in S$ is an extreme point of S if $\notin hull(S - P)$. In general, if S is finite, then $hull(S)$ is a convex polygon, and the extreme points of S are the corners of this polygon. The edges of this polygon are referred to as the edges of the $hull(S)$.

A supporting hyperplane is a another geometric concept. A hyperplane divides a space into two half-spaces. A hyperplane is said to support a set S in Euclidean space $\Re^K$ if it meets the following conditions:

- S is entirely contained in one of the two closed half-spaces of the hyperplane, and
- S has at least one point on the hyperplane.

In addition, if the dimension of the supporting line is higher than three, the related relationship can be written down as

$$S = \left(x \in \Re^K \middle| \sum_{j=1}^{K} \alpha_j x_j = b \right) \tag{9}$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_K]$ denotes a unit vector, $\mathbf{x} = [x_1, \ldots, x_K]$ is an arbitrary point and b assumes any arbitrary real value.

$$S^+ = \left(x \in \Re^K \middle| \sum_{j=1}^{K} \alpha_j x_j \geq b \right) \tag{10}$$

$$S^- = \left(x \in \Re^K \middle| \sum_{j=1}^{K} \alpha_j x_j \leq b \right) \tag{11}$$

In case when the following conditions are satisfied

$$S \bigcap P \neq \phi \quad \text{and} \quad P \subset S^+ \quad \text{or} \quad P \subset S^-, \tag{12}$$

it say that the supporting hyperplane S supports set P.

Using this definition, the reformulation of a convex hull called $conv(P)$, can be expressed as follows:

$$conv(P) = \bigcap_{S^{+}:uppersupportinghyperplane} S^{+} \tag{13}$$

$$conv(P) = \bigcap_{S^{-}:lowersupportinghyperplane} S^{-} \tag{14}$$

2.4.2 Beneath-Beyond Algorithm

This algorithm incrementally builds up the convex hull by keeping track of the current convex hull, P_i using an incidence graph. The Beneath-Beyond algorithm consists of the following steps [20]:

Step 1: Select and sort points along one direction, say x_1. Let $s = P_0, P_1, \ldots, P_{n-1}$ be input points after sorting. Process the points in an increasing order.

Step 2: Take the first n points, which define a facet as the initial hull.

Step 3: Let P_i be the point to be added to the hull at the ith stage. Let $P_i = conv(P_0, P_1, \ldots, P_{i-1})$ be the convex hull polytope built so far. This step includes two kinds of hull updates:

(a) A pyramidal update is done when $P_i \notin aff(P_0, P_1, \ldots, P_{i-1})$—when P_i is not on the hyperplane defined by the current hull. A pyramidal update consists of adding a new node representing P_i to the incidence graph and connecting this node to all existing hull vertices by new edges.

(b) A non-pyramidal update is done when the above condition is not met, i.e. P_i is in the affine subspace defined by the current convex hull. In this case, faces that are visible from P_i are removed and new facets are created.

2.5 A Convex Hull-Based Regression

In regression, deviations between observed and estimated values are assumed to be due to the random errors. Regression analysis is one of commonly encountered approaches in describing relationships among the analyzed data. The regression models explain dependencies between independent and dependent variables. The variables, which are used to explain the other variable(s) are called explanatory ones [21, 22].

Although conventional regression has been applied to various applications, problems may arise when they were encountered vague relationships between input and output variables in which cases there assumptions made for regression models are not valid any longer. This situation becomes a major reason behind a lack of relevance of regression models [23].

Recall that a standard numeric linear regression model comes in the following form:

$$Y = A_0 + A_1 x_1 + \cdots + A_K x_K. \tag{15}$$

As an interesting and useful extension, Tanaka et al. introduced an enhancement of the regression model by accommodating fuzzy sets thus giving rise to the term of fuzzy regression or possibilistic regression [24]. The models of this category reflect the fuzzy set based nature of relationships between the dependent and independent variables. The upper and lower regression boundaries in the fuzzy regression are used to quantify the fuzzy distribution of the output values.

As an alternative to the fuzzy specification, an inexact relationship among those dependent and independent variables can be represented via fuzzy linear regression expressed in the following form:

$$\tilde{Y} = \tilde{A}_0 x_0 + \tilde{A}_1 x_1 + \cdots + \tilde{A}_K x_K = \tilde{\mathbf{A}} \boldsymbol{x}^t \tag{16}$$

where $\mathbf{x} = [x_0, x_1, \ldots, x_K]$ is a vector of independent variables with $x_0 = 1$; $\tilde{\mathbf{A}} = [\tilde{A}_0, \tilde{A}_1, \ldots, \tilde{A}_K]$ is a vector of fuzzy coefficients represented in the form of symmetric triangular fuzzy numbers and denoted by $\tilde{A}_j = (\alpha_j, c_j)$ with membership function described as follows:

$$u_{\tilde{A}}(a_j) = \begin{cases} 1 - \frac{|\alpha_j - a_j|}{c_j}, & c_j \neq 0, \alpha_j - c_j \leq a_j \leq \alpha_j + c_j, \\ 1, & c_j = 0, \alpha_j = a_j, \\ 0, & otherwise, \end{cases} \tag{17}$$

where α_j and c_j are the central value and the spread of the triangular fuzzy number, respectively.

From the computational perspective, the estimation of the membership functions of the fuzzy parameters of the regression is associated with a certain problem of Linear Programming (LP) [21].

Given the notation used above, Eq. (16) can be rewritten as follows

$$\tilde{Y}_i = (\alpha_0, c_0) + (\alpha_1, c_1) x_1 + (\alpha_2, c_2) x_2 + \cdots + (\alpha_K, c_K) x_K. \tag{18}$$

where α_j and $c_j (j = 1, 2, \ldots, K)$ are the center and the spread of the predicted interval of $\tilde{A}_j$, respectively.

The weakness of the implementation of the multidimensional fuzzy linear regression can be alleviated by incorporating the convex hull approach [7, 25].

In the introduced modification, the construction of vertices of the convex hull becomes realized in real-time by using related points (convex points) of the graph. Furthermore, Ramli et al. stated that the real-time implementation of the method has to deal with a large number of samples (data). Therefore, each particular analyzed sample stands for a convex point and is possibly selected as a convex hull vertex. Some edges connecting the vertices need to be re-constructed as well [26].

Let us recall that the main purpose of fuzzy linear regression is to form the upper and lower bounds of the linear regression model. Both the upper line Y^U and lower line Y^L of the fuzzy linear regression are expressed in the form:

$$Y^U = \{A_0 + A_1x_1 + \cdots + A_Kx_K\}^U\colon \left\{\boldsymbol{A}\boldsymbol{x}_i^t\right\}^U = \boldsymbol{\alpha}\boldsymbol{x}_i^t + \boldsymbol{c}\left|\boldsymbol{x}_i^t\right| \tag{19}$$

$$Y^L = \{A_0 + A_1x_1 + \cdots + A_Kx_K\}^L : \{\boldsymbol{A}\boldsymbol{x}_i^t\}^L = \boldsymbol{\alpha}\boldsymbol{x}_i^t - \boldsymbol{c}|\boldsymbol{x}_i^t| \tag{20}$$

By using Eqs. (19) and (20), the problem was converted to a general fuzzy regression that is similar to the one shown below:

1. Evaluation (objective) function

$$\min_{\boldsymbol{\alpha},\mathbf{c}} \sum_{i=1}^{n} \sum_{j=2}^{K} c_j |P_{ij}|. \tag{21}$$

2. Constraints

$$P_{i1} \in Y_i \Leftrightarrow \begin{cases} P_{i1} \le \alpha_0 + c_0 + \sum\limits_{j=2}^{K} \alpha_j P_{ij} + \sum\limits_{j=2}^{K} c_j |P_{ij}| \\ P_{i1} \ge \alpha_0 - c_0 + \sum\limits_{j=2}^{K} \alpha_j P_{ij} - \sum\limits_{j=2}^{K} c_j |P_{ij}| \\ \qquad (i = 1, \ldots, n). \end{cases} \tag{22}$$

The above expression can be further rewritten as follows:

$$\begin{aligned} Y^U &= \{Y_i^U | i = 1, \ldots, n\} \\ Y^L &= \{Y_i^L | i = 1, \ldots, n\} \end{aligned} \tag{23}$$

Here also arrive at the following simple relations for P_{i1}

$$P_{i1} \le Y_i^U, \quad P_{i1} \ge Y_i^L \quad (i = 1, \ldots, n) \tag{24}$$

It is well known that any discrete topology is a topology which is formed by a collection of subsets of a topological space χ and the discrete metric ρ on χ is defined as

$$\rho(x,y)\begin{matrix} 1 & \textit{if } x \neq y \\ 0 & \textit{if } x = y \end{matrix} \tag{25}$$

for any $x, y \in X$. In this case, (X, ρ) is called a discrete metric space or a space of isolated points. According to the definition of discrete topology, expression (24) is rewritten as follows:

$$\begin{aligned} S(Y^U) &= \sum_{j=1}^{K} \{Y_j P_{ij}\}^U \geq 0 \\ S(Y^L) &= \sum_{j=1}^{K} \{Y_j P_{ij}\}^L \leq 0 \end{aligned} \tag{26}$$

where assume that $P_{i1} = 1$.

This formula corresponds with the definition of the support hyperplane. Under the consideration of the range of

$$S \bigcap P \neq \phi \quad \text{and} \quad P \subset S^+ \quad \text{or} \quad P \subset S^-, \tag{27}$$

the following relationship is valid:

$$\bigcap S(Y^U) = \bigcap S(Y^L). \tag{28}$$

This is explained by the fact that regression formula Y^U and Y^L are formed by vertices of a convex hull. Therefore, it is apparent that the constructed convex hull polygon or more specifically, its vertices clearly define the discussed constraints of fuzzy mathematical programming, becomes more reliable as well as significant for the subsequent processes.

Recall that the convex hull of a set S of points while $hull(S)$ is defined to be a minimum convex set containing S. A point $P \in S$ is an extreme point of S if $P \notin hull(S - P)$. Hence P denotes the set of points (input samples) and P_C is the set of vertices of the convex hull where $P_C \in P$. Therefore, the convex hull has to satisfy the following relationship:

$$conv(P) = conv(P_C) \tag{29}$$

Introduce here the following set

$$P_C = \{x_{Cl} \in \Re^K | l = 1, \ldots, m\} \subseteq P \tag{30}$$

where m is the number of vertices of the convex hull. Plugging this relationship into (22), at the following constraints was arrived.

$$P_{i1} \in Y_i \Leftrightarrow \begin{cases} P_{i1} \leq \alpha_0 + c_0 + \sum\limits_{j=2}^{K} \alpha_j P_{ij} + \sum\limits_{j=2}^{K} c_j |P_{ij}| \\ P_{i1} \geq \alpha_0 - c_0 + \sum\limits_{j=2}^{K} \alpha_j P_{ij} - \sum\limits_{j=2}^{K} c_j |P_{ij}| \\ (i = 1, \ldots, m). \end{cases} \tag{31}$$

In virtue of Eq. (31), the constraints of the LP of the fuzzy linear regression can be written in the following manner:

$$y_i \in Y_i \Leftrightarrow \begin{cases} y_i \leq \alpha \boldsymbol{x}_i^t + \boldsymbol{c} |\boldsymbol{x}_i^t| \\ y_i \geq \alpha \boldsymbol{x}_i^t - \boldsymbol{c} |\boldsymbol{x}_i^t| \\ (i = 1, \ldots, m). \end{cases} \tag{32}$$

Moreover, in order to form a suitable regression model based on the constructed convex hull, the connected vertex points are used as the constraints in the LP formulation of the fuzzy linear regression. Considering this process, the use of the limited number of selected vertices contributes to the minimized computing complexity associated with the model [1].

3 A Real-Time Granular Based Fuzzy Regression Models with a Convex Hull Implementation

In general, there are four major components of this proposed approach includes genetically-guided clustering, sub-convex hull construction process, main convex hull construction process and fuzzy regression solution. The description of related components is shown in Table 1.

Furthermore, Fig. 1 shows the synopsis of the entire processes where there are examples of four clustered sample of data (clustered feature vectors). In addition, this clustered feature vectors were representing information granules. Sub-convex hull were built for each of clustered feature vectors and based on Fig. 1, constructed of sub-convex hulls are clearly defined. Consequently, highlighted also a main convex hull which was constructed depending on initially build of sub-convex hulls. Therefore, this solution will covers entire clustered samples of data or in other words, this proposed approach might consider for producing optimum regression results.

In order to make clearly understand of the proposed approach, the flow of the overall processing is presented, see Fig. 2. Some selected samples of granular data were load into the system. Then, GA-FCM is used for assigning relevance number of granules. Additionally, GA-FCM has been selected in this process because the ability to clearly define as well as separate the raw samples into associated granule. Even though GA-FCM could be required some additional processing time comparing with conventional FCM, the accurate result of produced classes are achievable.

Table 1 A description of the main components of the proposed approach

No.	Component	Involved algorithm/ processes	Description
1.	Genetically-guided clustering	GA-FCM algorithm	The used of GA-FCM algorithm for identify appropriate clusters which were representing information granules
2.	Sub convex hull construction	Beneath-Beyond algorithm	Build a sub convex hull polygon for each identified cluster. This process will be repeated until all identified clusters achieved. The number of constructed convex hull should be same with constructed clusters
3.	Convex hull construction	Beneath-Beyond algorithm	Build a convex hull polygon, which covers the whole constructed sub convex hull polygon
4.	Fuzzy regression solution	LP formulation for fuzzy	Used convex hull vertices in LP formulation of regression formulation fuzzy regression for producing optimal models

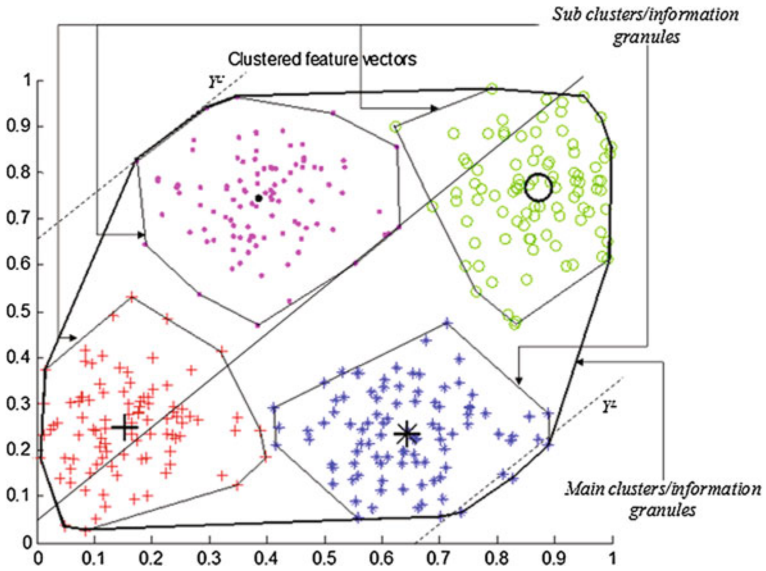


Fig. 1 An illustration of constructed sub-clusters and a main cluster

The following process involving convex hull construction where Beneath-Beyond algorithm has been selected here. During this process, the outer points of each constructed granule is completely identified. The selected outer points were connected each other for producing particular edges. The combination of connected edges will produce a convex hull polygon. In this situation, the produced convex hull

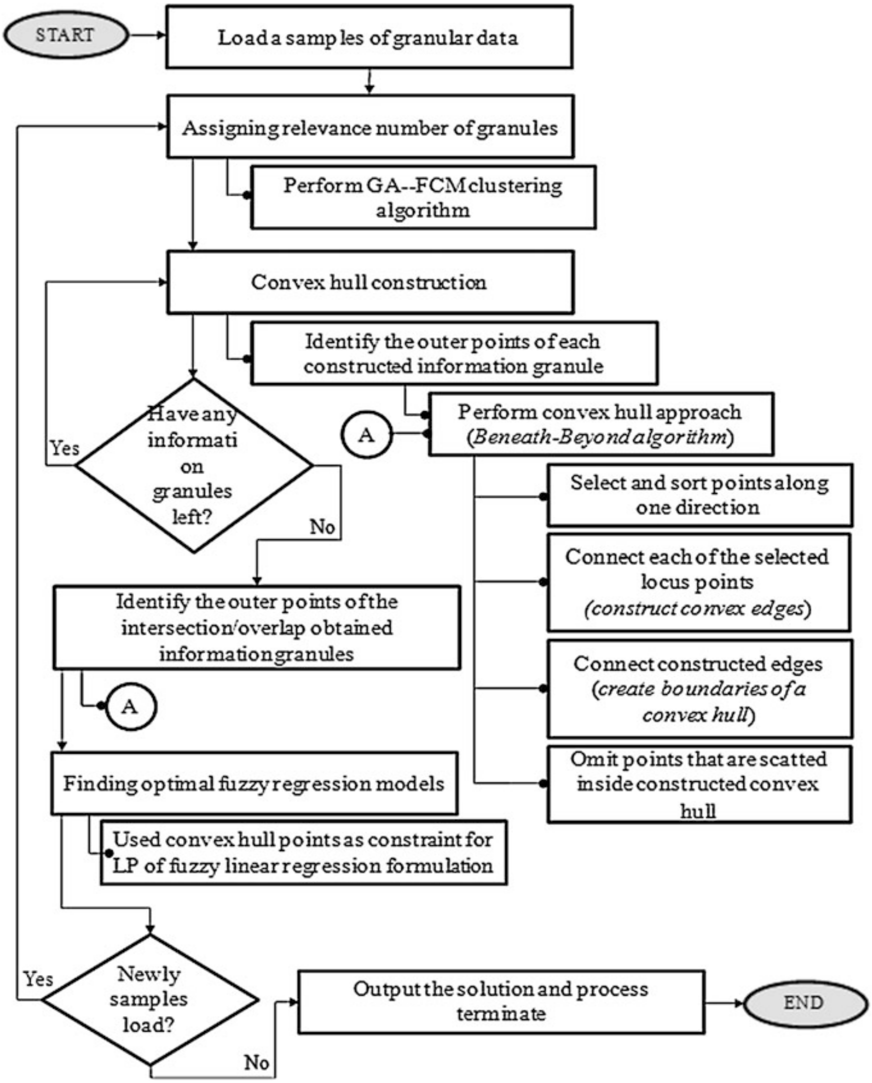

START
Load a samples of granular data
Assigning relevance number of granules
Perform GA--FCM clustering algorithm
Convex hull construction
Identify the outer points of each constructed information granule
Have any information granules left?
Yes
No
A
Perform convex hull approach (Beneath-Beyond algorithm)
Select and sort points along one direction
Connect each of the selected locus points (construct convex edges)
Connect constructed edges (create boundaries of a convex hull)
Omit points that are scatted inside constructed convex hull
Identify the outer points of the intersection/overlap obtained information granules
A
Finding optimal fuzzy regression models
Used convex hull points as constraint for LP of fuzzy linear regression formulation
Newly samples load?
Yes
No
Output the solution and process terminate
END

Fig. 2 A general flow of processing

categorized as sub-convex hull. Furthermore, these sub-processes will be iterated until desired point of data are classified under appropriate information granules.

The next sub-process is focusing on construction of main convex hull. This task will concentrate on finding the outer points which representing vertices of constructed sub-convex hulls by taking account of whole constructed sub-convex hull as once.

In the end, the final step consists of finding an optimal fuzzy regression model with utilization of main convex hull points or vertices. At this point, an optimal fuzzy regression models will be produced and process will be also terminated if the final group of data samples is fully arrived into the proposed approach and completely processed.

As a summarization of this part, some iterations of the overall procedure considering that more data become available in the future being completed. Say, new samples are provided within a certain time interval, e.g., they could be arrived every 10 s. Related to the comments made above, it becomes apparent that the quality of granular based fuzzy regression model can be improved by the hybrid combination of GA-FCM algorithm with convex hull-based fuzzy regression approach. The quality refers to the computing time as well as the overall computational complexity.

All in all, it do not have to consider the complete feature vectors for building regression models; just utilize the selected vertices, which are used for the construction of the convex hull. As mentioned earlier, these selected vertices come from a sub-convex hull, which represents appropriate information granules..

Therefore, this situation will lead to the decrease of the computation load. On the other hand, related to the computational complexity factor for the subsequent iteration, it will only consider the newly added samples of data together with the selected vertices of the previous convex hull (main constructed convex hull polygon). For that reason, this computing scenario will reduce the computational complexity because of the lower number of the feature vectors used in the subsequent processing of regression models.

4 A Numerical Example and Performance Analysis

A simple numerical example presented here, quantifies the efficiency of the proposed approach in the implementation of real-time granular based fuzzy regression.

As a guidance of this simulation example, Fig. 3 shows an illustration of a real-time reconstruction of a fuzzy classification analysis that involves a dynamic record/database. For instance, each of the iterations may have had the same amount of newly arrived data. As mentioned earlier, the amount of data increased as time progressed. Note that the initial group of samples was taken as the input for the first iteration process. It can see here the increase in the volume of data with the real-time arrival of new data.

Before going further into this precious section, affirmed here that, computer specification which has been used to perform the whole processes. The specifications of machine are; a personal notebook PC with Intel(R) Pentium CORE(TM) Duo 2 CPU (2.00 GHz) processors combined with 2 GB DDR2 type of RAM. Moreover, Windows Vista Business Edition (32 bit) was an operating system installed into this machine.

Based on Fig. 3, assume that an initial group of samples consists of 100 data of the well-known Iris data set [27]. Considering a distribution of these data,

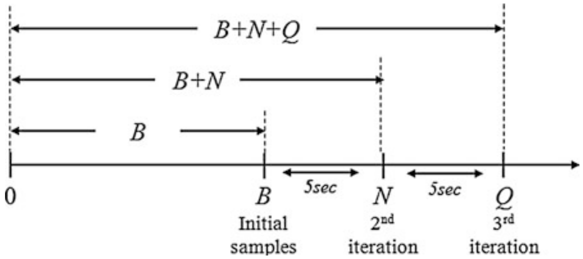


Fig. 3 An illustration of an increasing record/database along with time consumption

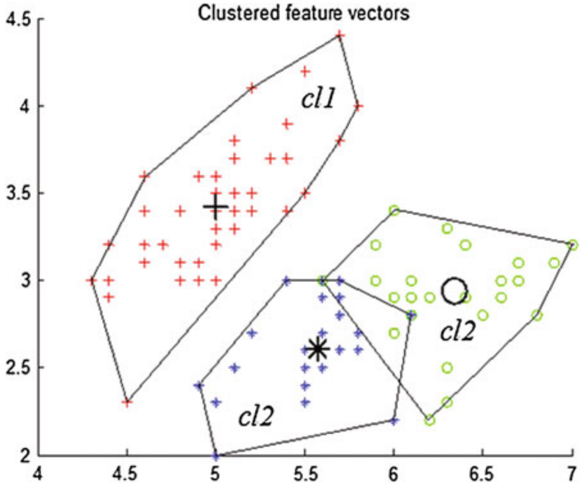


Fig. 4 Obtained clusters and constructed sub convex hulls for initial samples of data

constructed sub-convex hull polygons, which become the boundary of each identified cluster (or information granule) were successfully completed, see Fig. 4.

Referring to the figure highlighted (Fig. 4), there are 3 constructed sub-convex hulls called $cl1$, $cl2$ and $cl3$. Table 2 covers the details of all clusters.

Next, a main convex hull which covers those sub-convex hulls has been constructed and among 22 of total selected clustered feature vectors (or loci points) as stated in Table 2, only 11 points were selected as convex hull vertices, see Fig. 5. In addition, these selected vertices are located as the outside points of the constructed clusters. By solving the associated LP problem that considered these selected vertices as a part of the constraint portion standing in the problem, we obtained the optimal regression coefficients, see below. In addition, $h = 0.05$ has been selected to express goodness of fit or compatibility of data and the regression model

$$y = (2.071, 0.163) + (0.612, 0.096)C1 + (0.639, 0.075)C2 - (0.412, 0.000)C3$$

Table 2 Details of the obtained cluster along with the number of selected vertices for initial group of data samples

No.	Obtained clusters	Selected vertices
1.	Cluster 1 (*cl*1)	9
2.	Cluster 2 (*cl*2)	7
3.	Cluster 3 (*cl*3)	6

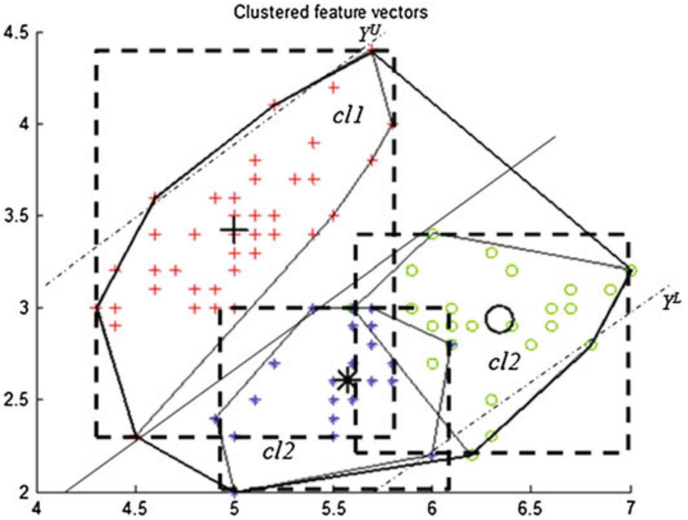


Fig. 5 Constructed of main convex hulls for initial samples of data

where

*C*1 input variable for Sepal Length,
*C*2 input variable for Sepal Width, and
*C*3 input variable for Petal Length,

with

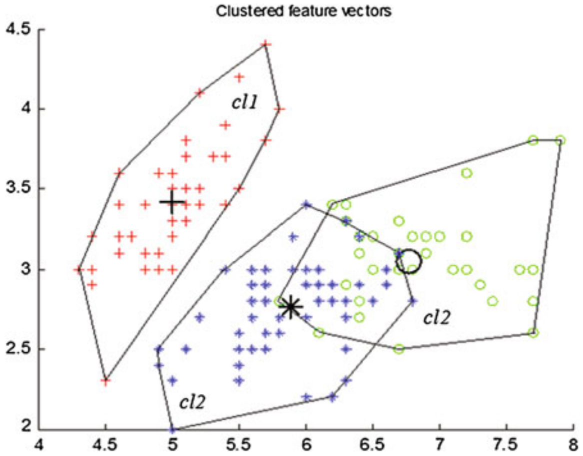
Constant value = 2.071, spread = 0.163,
Coefficient of Sepal Length = 0.612, spread = 0.096,
Coefficient of Sepal Width = 0.639, spread = 0.075, and
Coefficient of Petal Length = 0.412, spread = 0.000.

To deal with a real-time scenario, a group of samples taken from the same data set, which consists of 50 patterns has been added into previously selected patterns. In this case, assume that an iteration process has been completed. Table 3 shows the details of each sub-convex hull for initial group together with newly added data samples and Fig. 6 illustrate this related outcome.

Table 3 Detailed description of the clusters and the number of selected vertices for initial group together with newly added data samples

No.	Obtained clusters	Selected vertices
1.	Cluster 1 (*cl*1)	9
2.	Cluster 2 (*cl*2)	10
3.	Cluster 3 (*cl*3)	7

Fig. 6 Obtained clusters and constructed sub convex hulls for initial together with the newly added samples of data



The total number of selected vertices for this newly data volume is 26 and out of them, the main constructed convex hull only used 10 vertices, refer to Table 3. Finally, the obtained fuzzy regression model comes in the form;

$$y = (1.855, 0.173) + (0.651, 0.102)C1 + (0.709, 0.095)C2 - (0.556, 0.000)C3$$

where

$C1$ input variable for Sepal Length,
$C2$ input variable for Sepal Width, and
$C3$ input variable for Petal Length,

with

Constant value = 1.855, spread = 0.173,
Coefficient of Sepal Length = 0.651, spread = 0.102,
Coefficient of Sepal Width = 0.7099, spread = 0.095, and
Coefficient of Petal Length = 0.556, spread = 0.000,
while Fig. 7 shows the clustered feature vectors.

As early stated in the initial part of this research, one of main contribution towards this research is related with processing time factor. Consequently, the details of recoded time-length can be found in Table 4. In addition, this table also

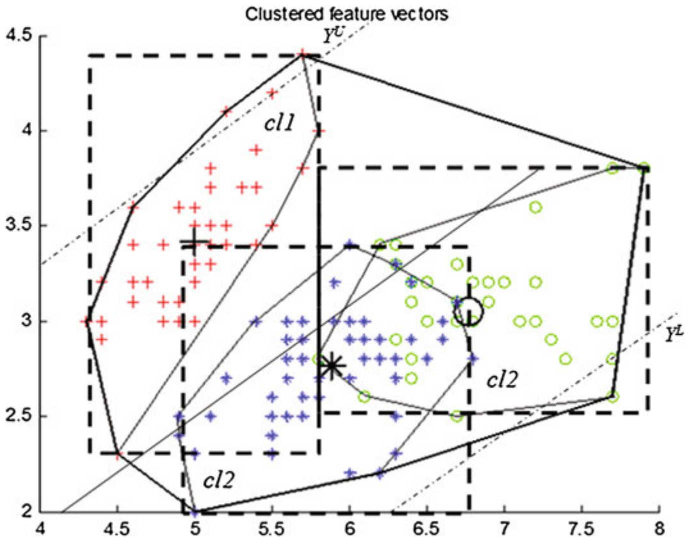


Fig. 7 Construction of main convex hulls for initial configuration together with newly added samples of data

Table 4 Granules-based fuzzy regression performance: details

Approach	Cycle (*Iteration*)	Selected feature vectors*	Time-length (s)
FCM classification with conventional regression	1st cycle	(22) [22]	01.42
	2nd cycle	(26) [26]	02.05
GA-FCM classification with conventional regression	1st cycle	(22) [22]	01.63
	2nd cycle	(26) [26]	02.24
Proposed of granular-based fuzzy regression approach	1st cycle	(22) [11]	00.28
	2nd cycle	(26) [10]	00.37

*(Total number of vertices—sub convex hull), [Total number of vertices—main convex hull]; ()- Represents equation; []- Represents reference

shows some related time instance which purposely for producing appropriate fuzzy regression models for identified information granules base on several conventional approaches. In this situation, the same samples of data were used while FCM as well as GA-FCM (both purposely for obtaining information granules class) together with conventional regression approach have been implemented accordingly.

It can see here, time-length recorded for initial samples of data (first cycle) is only 00.28 s and for the second following cycle is only needs 00.09 s additional time-length which becomes 00.37 s in total. Comparing with both combination of FCM with conventional fuzzy regression as well as GA-FCM with conventional fuzzy regression, notice that, the proposed approach looks more significant especially in term of time consumption. In addition, both of these combinations approach are likely not too much different particularly related to the time expenditure point of view and it can be realized here, that although some number of data samples are added together with initial group of data samples, the overall time consumption as well as computational complexity can be extremely decreased.

As previously discussed in the early portion of this section, the proposed approach can shorten overall time length due to the reused of produced sub as well as main convex hull polygon. Shown here also, deployment of FCM and GA-FCM with conventional regression approach, as tabled results, both of these combination have to consider all analyzed data for the first cycle and reconsider them again plus with newly arrived data for the second cycle, see Table 4. This situation requires additional time-length and computational complexity might be increased.

On the other hand, focusing to the accuracy factor of the produced regression models employing the proposed approach, noticed that those constructed models are likely similar comparing with the models which was generated through utilization of both FCM as well as GA-FCM classification approach combined with conventional regression approach. Additionally, the differences range of desire constant, coefficient and spread values are between 0.006. Therefore, it can be concluded that, the precision level of obtained fuzzy regression models with the use of the proposed approach is greatly accepted.

In summary, it can be highlighted here that, the proposed of granular-based fuzzy regression reaches the best performance for real-time data processing.

5 Conclusion and Future Works

In this chapter, an enhancement of the IDA tool of fuzzy regression completed in the presence of information granules have been proposed. Generally, the proposed approach first constructs a limited number of information granules and afterwards the resulting granules are processed by running the convex hull-based regression [6]. In this way, it have realized a new idea of real-time granular based fuzzy regression models being viewed as a modeling alternative to deal with real-world regression problems.

It is shown that information granules are formed as a result of running the genetic version of the FCM called GA-FCM algorithm [3]. Basically, there are two parts of related process, which utilize the convex hull approach or specifically Beneath-Beyond algorithm; constructing sub-convex hull for each identified clusters (or information granules) and building a main convex hull polygon which covers all constructed sub-convex hulls. In other word, the main convex hull is completed depending upon the outer plots of the constructed clusters (or information granules). Additionally, the sequential flow of processing was carried out to deal with dynamically increasing size of the data.

Based on the experimental developments, one could note that, this approach becomes a suitable design alternative especially when solving real-time fuzzy regression problems with information granules. It works efficiently for real-time data analysis given the reduced processing time as well as the associated computational complexity.

This proposed approach can be applied to real-time fuzzy regression problems in large-scale systems present in real-world scenario especially involving granular computing situation. In addition, each of the implemented phases, especially GA-FCM process and both sub and main convex hull construction processes have their own features in facing with dynamically changes of samples volume within a certain time interval. As a result, this enhancement (or hybrid combination) provides an efficient platform for regression purposes. Although in this paper it dealt with small data sets (and this was done for illustrative purposes), it is worth noting that method scales up quite easily.

In further studies, it plan to expand the proposed approach by incorporating some other technologies of soft computing and swarm intelligence techniques such particle swarm optimization (PSO) or ant colony optimization (ACO).

Acknowledgments The first author was worked with Universiti Tun Hussion Onn Malaysia, MALAYSIA and enrolled as PhD Candidate at Graduate School of Information, Production and Systems (IPS), Waseda University, Fukuoka, JAPAN.

References

1. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Kluwer Academic Publishers, Dordrecht (2003)
2. Shifei, D., Li, X., Hong, Z., Liwen, Z.: Research and progress of cluster algorithms based on granular computing. Int. J. Digit. Content Technol. Appl. **4**(5), 96–104 (2010)
3. Snijders, C., Matzat, U., Reips, U.-D.: 'Big data': Big gaps of knowledge in the field of internet. Int. J. Internet Sci. **7**, 1–5 (2012)
4. Hall, L.O., Ozyurt, I.B., Bezdek, J.C.: Clustering with a genetically optimized approach. IEEE Trans. Evol. Comput. **3**(2), 103–112 (1999)
5. Bargiela, A., Pedrycz, W.: Toward a theory of granular computing for human centered information processing. IEEE Trans. Fuzzy Syst. **16**(16), 320–330 (2008)
6. Chen, B., Tai, P.C., Harrison, R., Pan, Y.: FIK model: Novel efficient granular computing model for protein sequence motifs and structure information discovery. In: 6th IEEE International Symposium on BioInformatics and BioEngineering (BIBE 2006), Arlington, Virginia, pp. 20–26 (2006)
7. Ramli, A.A., Watada, J., Pedrycz, W.: Real-time fuzzy regression analysis: A convex hull approach. Eur. J. Oper. Res. **210**(3), 606–617 (2011)
8. Ramli A.A., Watada, J.: New perspectives of fuzzy performance assessment of manufacturing enterprises. In: The 5th International. Conference on Intelligent Manufacturing and Logistics Systems (IML 2009), Waseda University, Kitakyushu, Japan, pp. 16–18 (2009)
9. Yu, P.-S., Chena, S.-T., Changa, I.-F.: Support vector regression for real-time flood stage forecasting. J. Hydrol. **328**(3–4), 704–716 (2006)
10. Wang, W., Chena, S., Qu, G.: Incident detection algorithm based on partial least squares regression. Transp. Res. Part C: Emerg. Technol. **16**(1), 54–70 (2008)
11. Pedrycz, W., Vulcovich, G.: Representation and propagation of information granules in rule-based computing. J. Adv. Comput. Intell. Intell. Inf. **4**(1), 102–110 (2000)

12. Hoppner F., Klawonn, F.: Systems of information granules. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Handbook of Granular Computing. John Wiley & Sons Ltd, Chichester (2008). doi:10.1002/9780470724163.ch9
13. Chen, B., Hu, J., Duan, L., Gu, Y.: Network administrator assistance system based on fuzzy C-means analysis. J. Adv. Comput. Intell. Intell. Inf. **13**(2), 91–96 (2009)
14. Nascimento, S., Mirkin, B., Moura-Pires, F.: A fuzzy clustering model of data and fuzzy C-means. In: IEEE Conference on Fuzzy Systems (FUZZ-IEEE2000), San Antonio, Texas, USA, pp. 302–307 (2000)
15. Alata, M., Molhim, M., Ramini, A.: Optimizing of fuzzy C-means clustering algorithm using GA. World Acad. Sci. Eng. Technol. 224–229 (2008)
16. Yabuuchi, Y., Watada, Y.: Possibilistic forecasting model and its application to analyze the economy in Japan. Lecture Notes in Computer Science, vol. 3215, pp. 151–158. Springer, Berlin, Heidelberg (2004)
17. Lin, H.J., Yang, F.W., Kao, Y.T.: An efficient GA-based clustering technique. Tamkang J. Sci. Eng. **8**(2), 113–122 (2005)
18. Wang, Y.: Fuzzy clustering analysis by using genetic algorithm. ICIC Express Lett. **2**(4), 331–337 (2008)
19. Emiris, Z.: A complete implementation for computing general dimensional convex hulls. Int. J. Comput. Geometry Appl. **8**(2), 223–249 (1998)
20. Barber, B., Dobki, D.P., Hupdanpaa, H.: The quickhull algorithm for convex hull. ACM Trans. Math. Softw. **22**(4), 469–483 (1996)
21. Wang, H.-F., Tsaur, R.-C.: Insight of a possibilistic regression model. Fuzzy Sets Syst. **112**(3), 355–369 (2000)
22. Watada, J., Pedrycz, W.: A possibilistic regression approach to acquisition of linguistic rules. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Handbook on granular commutation, pp. 719–740. John Wiley and Sons Ltd., New York (2008)
23. Ramli, A.A., Watada, J., Pedrycz, W.: An efficient solution of real-time fuzzy regression analysis to information granules problem. J. Adv. Comput. Intell. Intell. Inf. (JACIII) **16**(2), 199–209 (2012)
24. Tanaka, H., Uejima, S., Asai, K.: Linear regression analysis with fuzzy model. IEEE Trans. Syst. Man Cybern. **12**(6), 903–907 (1982)
25. Ramli, A.A., Watada, J., Pedrycz, W. Real-time fuzzy switching regression analysis: A convex hull approach. In: 11th International Conference on Information Integration and Web-based Applications and Services (iiWAS2009), Kuala Lumpur, Malaysia, pp. 284–291 (2009)
26. Ramli, A.A., Watada, J., Pedrycz, W.: A combination of genetic algorithm-based fuzzy C-means with a convex hull-based regression for real-time fuzzy switching regression analysis: Application to industrial intelligent data analysis. IEEJ Transactions on Electr. Electron. Eng. **9**(1), 71–82 (2014)
27. Frank A., Asuncion, A.: UCI machine learning repository. University of California, School of Information and Computer Science, Irvine, CA. http://archive.ics.uci.edu/ml (2010)

How to Understand Connections Based on Big Data: From Cliques to Flexible Granules

Ali Jalal-Kamali, M. Shahriar Hossain and Vladik Kreinovich

Abstract One of the main objectives of science and engineering is to predict the future state of the world—and to come up with actions which will lead to the most favorable outcome. To be able to do that, we need to have a quantitative model describing how the values of the desired quantities change—and for that, we need to know which factors influence this change. Usually, these factors are selected by using traditional statistical techniques, but with the current drastic increase in the amount of available data—known as the advent of *big data*—the traditional techniques are no longer feasible. A successful semi-heuristic method has been proposed to detect true connections in the presence of big data. However, this method has its limitations. The first limitation is that this method is heuristic—its main justifications are common sense and the fact that in several practical problems, this method was reasonably successful. The second limitation is that this heuristic method is based on using "crisp" granules (clusters), while in reality, the corresponding granules are flexible ("fuzzy"). In this chapter, we explain how the known semi-heuristic method can be justified in statistical terms, and we also show how the ideas behind this justification enable us to improve the known method by taking granule flexibility into account.

Keywords Connections · Big data · Flexible granules · Intelligence analysis · Biomedical publications

A. Jalal-Kamali · M. Shahriar Hossain · V. Kreinovich (✉)
Department of Computer Science, University of Texas at El Paso, El Paso, TX 79968, USA
e-mail: vladik@utep.edu

A. Jalal-Kamali
e-mail: ajalalkamali@miners.utep.edu

M. Shahriar Hossain
e-mail: mhossain@utep.edu

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_4,

1 Understanding Connections Based on Big Data: An Important Practical Problem

What are our main objectives? The role of science and engineering. We have preferences: we want tasty food, we want a comfortable environment, we want to stay healthy, etc. In general, we have many objectives. We are making individual and collective decisions so as to satisfy these objectives; to be more precise, we select actions which maximize our degree of satisfaction in these objectives.

To be able to select appropriate actions, we need to be able to predict the consequence of different actions. Crudely speaking, this is what we usually understand by *science*: we know the current state of the world, we describe what actions we plan to perform, and we want to predict the future state of the world.

Once we can do that, we need to select a sequence of actions which will be the most beneficial; crudely speaking, this is what we usually understand by *engineering*. For example:

- Science predicts what happens to a rocket if we launch it in a certain direction.
- Based on these predictions, we can solve an engineering problem—find in what direction we must launch a rocket so that it will, for example, reach the Moon.

While praising successes of science and engineering, we need to remember that these successes are based on understanding connections. In the last several centuries, science and engineering achieved many things—we have successfully overcome many diseases, we drastically increases the life expectancy, we reached the Moon. These successes are based on complex quantitative methods of modern science and engineering.

In spite of all these successes, in some areas—such as economics—we still do not have good predictive models. The reason is simple. In general, there are many factors which could potentially affect the desired values. In many physics problems, we have succeeded in pinpointing a few relevant factors—and showing that all other factors can be safely ignored. For example, the acceleration of a rocket is determined by the forces acting on this rocket—gravity and aerodynamic resistance. Once we know that the desired value depends on the few parameters, we can use experiments to find the exact quantitative form of this dependence.

In contrast, in economics, we cannot dismiss any of the factors. As a result, potentially, we have a function of very many variables. To describe such functions, we need a very large number of parameters—much more parameters than the number of data points.

In other words, to be able to build a successful quantitative model, we first need to understand with which quantities the desired quantity is connected—and with which it is not. In other words, understanding connections is an important prerequisite for successes of science and engineering.

This importance can be also illustrated on examples from medicine. For some diseases—like cholera or malaria—originally many factors were considered: for example, that malaria is caused by swampy air, etc. (not to count such weird hypothesis as witchcraft and divine punishment for sins). When many possible factors were considered, no easy model of these illnesses existed, and no good cure was known. Once the scientists succeeded in determining the unique factor determining each of these diseases—the corresponding bacteria—this opened the possibility for developing successful medicine.

In contrast, for many types of cancer, we still have too many possible factors—viruses, pollution, stress, genetic mutations, etc. As a result, for these cancers, we do not have a good cure.

How connections are determined now. Traditionally, connections are determined by statistical methods; see, e.g., [18]. We observe some relation between the two processes: e.g., we observe that patients getting a certain medicine tend to recover faster, that the two DNA samples match, etc. This may be a random coincidence. So, in order to check whether the observed relation is statistically significant, we compute the probability p that this observed relation can happen for two unrelated processes. If this probability is smaller than a certain threshold p_0 (called a *p-value*), we conclude that there is a statistically significant connection; if the probability p is larger than p_0, then we cannot make this conclusion. Usually, practitioners take $p_0 = 0.05$ or, sometimes, $p_0 = 0.01$.

The connection-building task has been used in a variety of contexts: entity networks [5, 8], image collections [6], cellular networks [2, 7], social networks [4], and document collections [8, 9, 11]. All these research efforts focus on finding connections between objects that are apparently disjoint. A solution to the connection building task generally depends on the commonality between some intermediaries to reach the target object. Swanson refers to the notion of neighboring commonality as complementary but disjoint (CBD) structures [20], whereby two arguments may exist separately that when considered together lead to new insights, but the objects exhibiting these two arguments are unaware of each other. The proposed solution to connection building in this chapter leverages a similar principle.

Enter big data. Modern technology has led to a drastic increase in the amount of possible observations—and in the number of parameters related to each observation that we can measure and record. In principle, with devices like Google Glass, we can record everything that we see—and more generally, everything that is happening in the world. The resulting amount of data is so huge that not only a single researcher cannot review all this data—even the existing computer algorithms cannot process all this data. This phenomenon is known as *big data*; see, e.g., [3, 14, 19].

Traditional methods do not work well for big data: formulation of the problem. In the traditional statistical approach, we made few observations, so observed connections were relatively rare. In the big data, we record so many parameters that everything appears connected.

For example, traditionally, when we had to rely on human witnesses, the fact that the victim and the suspect were seen together (or could be indirectly connected by a convincing chain of such seen-together events) was a strong argument for the suspect's guilt.

Nowadays, with numerous security cameras recording many moments of our lives—from walking the streets to attending football games on a stadium—there are so many pairs of people who happen to be together at the same time in the same place simply by accident, that it is extremely difficult to separate such random encounters from true connections.

So, for big data, we need new methods to find out which joint appearances correspond to true connections and which do not.

Technical challenges related to the use of big data. One of the main challenges in using big data is that, as we have just mentioned, the use of big data leads to the need for developing new data processing algorithms.

However, even in situations when we can use the traditional data processing algorithms, the need to apply these algorithms to big data often leads to technical challenges. For example, in many practical situations, data processing data starts by estimating the usual statistical characteristics such as covariances etc. The usual algorithms for computing these characteristic assume that the whole data set is placed in the computer memory—and algorithms whose running time is quadratic or even cubic in terms of the size of the data set are quite feasible. In contrast, for big data, the size of the data set exceeds the computer memory's ability—and for a data set consisting of billions of records, quadratic-time algorithms require 10^{18} computational steps—which is not very realistic, even on highly parallel computers; see, e.g., discussions in Sect. 3.

All this need to be taken into account when we process big data.

What we do in this chapter. Our main objective is to study how to detect true connections based on the big data.

- We start with describing the semi-heuristic methods which have been proposed for solving this problem, as described, e.g., in [8, 9].
- Then, we describe the limitations of the existing methods. Some of these limitations are related to the fact that the existing methods are based on using *crisp* granules (clusters), while real-life clusters are flexible ("fuzzy"); see, e.g., [15].
- Finally, we describe how these limitations can be overcome—in particular, how we can use flexible granules (clusters) to understand true connections based on the big data.

Two case studies. The existing method has been tested on two big-data situations.

First case study: intelligence analysis. The paper [8] deals with *intelligence analysis*. Specifically, we have a huge database of documents. Based on these documents, we need to detect possible true connections between adversaries. The existing documents provide only possible relation—e.g., if two names appear in the same document, this may be an indication that the two persons are connected. The document may combine the name of the person with the name of the hotel where this person stayed at a certain night—and if another document shows another person staying at the same hotel, this may be an indication of a true connection between them.

The mere fact that the two names appeared in the same document does not necessarily mean that these names are actually connected—for example, one of the authors (V.K.) was born in the same city of St. Petersburg, Russia, as Grigory Perelman (of the Poincaré conjecture fame), graduated from the same St. Petersburg University, shared the same PhD advisor—but he never met Perelman in person, so there is clearly no true direct connection. However, if there are many such connecting documents, it increases the probability that the two names are actually connected—and at some point, we should be able to conclude, with a reasonable confidence, that there is a true connection.

Second case study: biomedical publications. The paper [9] deals with biomedical publications. The field of biomedical research has become so specialized that it is no longer easy for a human specialist to trace all relevant papers—or even to find all relevant papers. Finding such relevant papers is extremely important because in many cases, by combining the ideas presented in related papers, we can come up with a synergistic effect of an even better cure. Here also, we have a huge database of documents—this time, of papers. Based on these documents, we want to find true connections between the papers.

Similar to the intelligence analysis case, we can come up with criteria of when two papers may be connected: e.g., if they share keywords or share references, etc. Based on this information, it is necessary to decide when the two papers are actually connected and when the seeming connection is accidental.

2 General Case: How to Describe Available Information

General situation. In general:

- We have a large set of *entities*: persons, locations, organizations, dates, etc. for the intelligence database, biomedical articles, etc.
- We also have a huge database of *features*: documents for the intelligence database, biomedical terms for the publications database, etc.—which enable us to relate some entities.

Based on this information, we have to decide which entities are actually connected and which are not.

Description of the available information. In general:

- we have entities e,
- we have features f, and we have *associations* between entities e and features f: for example,

– a name e is mentioned in the document f,
– a term f appears in a paper e, etc.

For some e and f, we may have several associations—e.g., the name e is mentioned several times in the document f, or the term f appears several times in the paper e.

Some other notations are as follows:

- we will denote the set of all entities by $\mathscr{E}$;
- we will denote the set of all features by $\mathscr{F}$; and
- for each e and f, we will denote the number of associations between e and f by $n_{e,f}$.

The total number of entities is equal to $|\mathscr{E}|$ and the total number of features is equal to $|\mathscr{F}|$. It is also useful to describe:

- for each feature f, the set $e(f) \stackrel{\text{def}}{=} \{e \in \mathscr{E} : n_{e,f} > 0\}$ of all entities associated with the feature f, and
- for each entity e, the set $f(e) \stackrel{\text{def}}{=} \{f \in \mathscr{F} : n_{e,f} > 0\}$ of all features associated with the entity e.

First step of the usual document analysis: describing the weight $V(e,f)$ of the feature f for the entity e. Based on information about associations between entities and features, we can decide which features are more important for a given entity and which are less important.

Intuitively, the larger the number of associations between the entity and the feature, the more confident we are that this association is meaningful—for example, one mention of a name in a document may be accidental, but if the same name appears several times, we become confident that this is a connection between the name and the document.

Similarly, the fewer entities are associated with the feature, the more confident we are that this association is meaningful. When two people are listed in the same document, then how meaningful is this association depends on how many other people are listed in this document. For example, if two people are listed in the same New York City phone book, a document that lists millions of other people, this does not mean much beyond the fact that they both live in New York City—and is clearly not an indication that there is a special connection between these two people. On the other hand, if two people are listed in the hotel bills issued on the same day by the same small bed-and-breakfast hotel, then there is a high possibility that they met each other—e.g., at breakfast.

Let us describe this qualitative idea in numerical terms. In situations like this, when we have several entities associated with a feature, a reasonable idea is to use the amount of information, i.e., the number of binary (“yes”–“no”) questions (bits) which are needed to find the desired entity.

In general, if we know that an unknown object belongs to the set consisting of N elements, then we can divide this set into two halves and, by asking a binary question, find out which half the desired object belongs to. After we receive a reply to the binary question, we know that the objects belongs to one of the corresponding halves. So, after we get the reply to the first binary question, we now have a set with $\frac{N}{2} = N \cdot 2^{-1}$ elements which is known to contain the unknown object. By asking the second binary question, we can again halve the resulting set; so, after we get answers to two binary questions, we have a set with $\frac{N}{4} = N \cdot 2^{-2}$ elements which contain the unknown object. After three binary questions, we get a set with $\frac{N}{8} = N \cdot 2^{-3}$ elements, etc. In general, after receiving answers to q binary questions, we get a set of $N \cdot 2^{-q}$ elements which contains the desired element. When we reach $N \cdot 2^{-q} = 1$, this means that this set consists of the single element—i.e., that we have pinpointed the desired alternative. Thus, for the case of N alternatives, the corresponding information (number of binary questions) can be determined from the equation $N \cdot 2^{-q} = 1$, and is, thus, equal to $q = \log_2(N)$.

Originally, we have $|\mathscr{E}|$ entities; the corresponding amount of information is equal to $\log_2(|\mathscr{E}|)$ bits. Once we know that an entity is associated with the feature f, we thus limit ourselves to $|e(f)|$ entities; in this case, the corresponding amount of information is equal to $\log_2(|e(f)|)$ bits. Thus, the very fact that the entity is associated with the feature f enables us to reduce the number of questions by the value

$$\log_2(|\mathscr{E}|) - \log_2(|e(f)|) = \log_2\left(\frac{|\mathscr{E}|}{|e(f)|}\right). \tag{1}$$

Similarly, the effect of multiple associations can be describe by counting how many additional binary questions we can afford and still keep an association with the desired entity. We start with $n_{e,f}$ mentions. Each binary question decreases this number by half; q questions decrease this amount to $n_{e,f} \cdot 2^{-q}$. As long as this remaining number is ≥ 1, we still have some association. The largest number q for which we can still get as association can thus be determined from the condition that $n_{e,f} \cdot 2^{-q} = 1$, and is, thus, equal to $q = \log_2(n_{e,f})$. To take into account the fact that we deal with *additional* questions, we usually add 1, ending up with $1 + \log_2(n_{e,f})$.

The overall importance of the feature f in entity e can be obtained if we multiply $\log_2\left(\frac{|\mathscr{E}|}{|e(f)|}\right)$ by the importance factor $1 + \log_2(n_{e,f})$, resulting in the product

$$I(e,f) \stackrel{\text{def}}{=} (1 + \log_2(n_{e,f})) \cdot \log_2\left(\frac{|\mathscr{E}|}{|e(f)|}\right). \tag{2}$$

This formula is one of the versions of *term frequency—inverse document frequency (tf-idf)* modeling; see, e.g., [12, 16].

For each entity e, we thus get the importance $I(e, f)$ of different features f. These values of importance are usually normalized, i.e., multiplied by a constant so that the mean square importance is equal to 1 (this is known as *cosine normalization*). As a result, we get the formula

$$V(e,f) = \frac{(1+\log_2(n_{e,f}))\cdot\log_2\left(\frac{|\mathscr{E}|}{|e(f)|}\right)}{\sqrt{\sum_{j\in f(e)}\left((1+\log_2(n_{e,j}))\cdot\log_2\left(\frac{|\mathscr{E}|}{|e(j)|}\right)\right)^2}}. \tag{3}$$

From weights to distance between entities. For each entity e, we have the weights $V(e, f)$ corresponding to different features f. Thus, as a measure of closeness between two entities e_1 and e_2, we can take the distance between the corresponding vectors $(V(e,f_1), V(e,f_2), \ldots)$.

In the usual Euclidean distance $d(a,b) = \sqrt{(a_1 - b_1)^2 + \cdots}$, we add the squares of the differences. Since each value $V(e, f)$ represents the number of bits, it makes more sense to take the actual differences—since each difference reflects the number of additional questions. Thus, we take

$$d(e_1, e_2) \stackrel{\text{def}}{=} \sum_{f\in\mathscr{F}} |V(e_1,f) - V(e_2,f)|. \tag{4}$$

This distance depends on the number of features: e.g., if, in addition to the documents, we store their copies, the distance increases by a factor of two. To avoid this dependence, the distance $d(e_1, e_2)$ is usually normalized to the interval [0,1]—by dividing by the largest possible value of this distance.

How can we estimate the largest possible value of this distance? In general, when we do not know the actual values a and b of two non-negative quantities, and we only know upper bounds $\overline{a}$ and $\overline{b}$ on these quantities, then the largest possible value of the difference $|a - b|$ is equal to $\max(\overline{a}, \overline{b})$. Indeed:

- if $\overline{a} \le \overline{b}$, then $|\overline{a} - \overline{b}| = \overline{b} - \overline{a} \le \overline{b}$ and thus, $|\overline{a} - \overline{b}| \le \max(\overline{a}, \overline{b})$;
- similarly, if $\overline{b} \le \overline{a}$, then $|\overline{a} - \overline{b}| = \overline{a} - \overline{b} \le \overline{a}$ and thus, $|\overline{a} - \overline{b}| \le \max(\overline{a}, \overline{b})$.

Thus, in both cases, we have $|\overline{a} - \overline{b}| \le \max(\overline{a}, \overline{b})$.

The bound $\max(\overline{a}, \overline{b})$ can be attained:

- if $\overline{a} \le \overline{b}$, then it is attained for $a = 0$ and $b = \overline{b}$;
- if $\overline{b} \le \overline{a}$, then it is attained for $a = \overline{a}$ and $b = 0$.

By applying this result to $\overline{a} = V(e_1,f)$ and $\overline{b} = V(e_2,f)$, we conclude that for each f, the maximum possible value of the difference

$$|V(e_1,f) - V(e_2,f)| \tag{5}$$

can be estimated as $\max(V(e_1,f), V(e_2,f))$. Therefore, the largest possible value of the sum $\sum_{f\in\mathscr{F}} |V(e_1,f) - V(e_2,f)|$ can be estimated as

$$\sum_{f \in \mathscr{F}} \max(V(e_1,f), V(e_2,f)). \tag{6}$$

By dividing $d(e_1, e_2)$ by this bound, we get the formula

$$D(e_1, e_2) \overset{\text{def}}{=} \frac{\sum_{f \in \mathscr{F}} |V(e_1,f) - V(e_2,f)|}{\sum_{f \in \mathscr{F}} \max(V(e_1,f), V(e_2,f))}. \tag{7}$$

This formula is known as the *Soergel distance*.

Comment. It is worth mentioning that the Soergel distance is a *metric*, in the sense that it is symmetric $D(e_1, e_2) = D(e_2, e_1)$ and satisfies the triangle inequality $D(e_1, e_3) \le D(e_1, e_2) + D(e_2, e_3)$.

Resulting description. As a result of the above preliminary analysis, we represent the given information as a *weighted graph*:

- in this graph, nodes (vertices) represent entities, i.e., the set of all the nodes is the set of all the entities $\mathscr{E}$;
- for each two entities (nodes) e_1 and e_2, we know the distance $D(e_1, e_2)$; in graph terms, this distance can be represented as the weight of the edge between e_1 and e_2.

3 A Known Semi-heuristic Method for Detecting True Connections Based on Big Data: A Brief Description

Direct and indirect connections. In some cases, we have a *direct* connection between the two objects—e.g., when two (or more) terrorist suspects meet together to plot future attacks.

Sometimes, the two suspects never (or rarely) meet in person, but they are plotting together via intermediaries—in this case, we have an *indirect* connection. In this case, we have a direct connection between the first suspect and the intermediary, and we have a direct connection between the intermediary and the second suspect—and we can use these two direct connections to make a conclusion that the two suspects are indirectly connected.

Detecting indirect connections is based on detecting direct ones. Because of this:

- we will first describe how direct connections are detected, and then
- we will describe how detected direct connections are combined to detect indirect connections.

From the original weighted graph to a simpler (non-weighted) one. In general, for every two nodes e_1 and e_2, we know the distance $D(e_1, e_2)$. The larger the distance, the less probable it is that the corresponding entities are actually connected.

- When the distance is very small, there is a high probability that the entities are connected. So, it is possible to conclude that the entities are connected if we need to make a definite decision about the connectivity.
- When the distance is close to 1, this probability becomes very small. So, we can conclude that the entities are *not* connected when a boolean decision about the connectivity is essential.

As we increase the distance from 0 to 1, there should be a point θ at which our decision changes from "connected" to "not connected". Once this threshold value θ is determined, we can then simplify the original weighted graph into a simplified non-weighted graph $\mathcal{G}$. In this simplified graph, the nodes (entities) e_1 and e_2 are connected by an edge if and only if $D(e_1, e_2) \leq \theta$.

Detecting direct connections: idea. As we have mentioned, if we have an edge between two entities e_1 and e_2, it is probable that there is an actual connection, but we cannot conclude this with confidence—since the edge may be caused by coincidence. If we also have a third entity e_3, and every two of the three entities e_1, e_2, and e_3 have an edge, then the probability that all the three edges are accidental is much smaller. As a result, our confidence that e_1 and e_2 are connected increases. Similarly, if there is a fourth entity e_4 and every two out of four entities have an edge, the probability increases.

In general, we may have ℓ entities e_1, e_2,..., e_ℓ for which every two entities have an edge. Such a set of nodes is known as an *ℓ-clique*. The larger ℓ, the higher our degree of confidence that e_1 and e_2 are actually connected. Thus, there is a threshold value k starting from which this confidence becomes so large that we can confidently conclude that e_1 and e_2 are actually connected.

This idea leads to the following algorithm for detecting direct connections.

Detecting direct connections: resulting method. We select a distance threshold $\theta \in (0, 1)$ and an integer k. We claim that two nodes e_1 and e_2 are actually directly connected in the graph $\mathcal{G}$ if in this graph, there is a k-clique containing both e_1 and e_2.

In other words, we claim that the entities e_1 and e_2 are directly connected if there exist edges e_3,...,e_k such that $D(e_i, e_j) \leq \theta$ for all i,$j \in \{1, 2, \ldots, \ell\}$.

Detecting a general connection: resulting method. A natural idea is to claim that the nodes e_1 and e_2 are actually connected if there is a chain of nodes $c_1 = e_1$, c_2,..., c_t, $c_{t+1} = e_2$ such that for every i, the nodes c_i and c_{i+1} are actually directly connected. This is equivalent to saying that in the graph $\mathcal{G}$, there is a chain of k-cliques G_1, G_2,..., G_t which connect e_1 and e_2 in the sense that:

- the first clique G_1 contains the node e_1,
- every two neighboring cliques have at least one common node, that is, $G_i \cap G_{i+1} \neq \emptyset$, and
- the last clique G_t contains the node e_2.

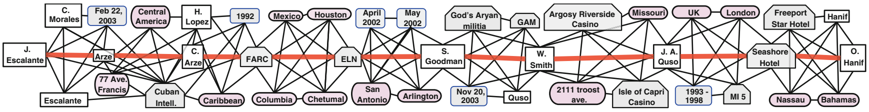


Fig. 1 Connection between the two suspects (Reproduced from [8])

How to select parameters of the method. The method described above used two parameters: θ and k. The values θ and k need to be determined empirically—e.g., by using examples where true connections are known and finding the values θ and k for which this method reproduces these known true connections as accurately as possible.

For example, for intelligence analysis [8], the values $\theta = 0.93$ and $k = 6$ lead to a good outcome; see Fig. 1.

How to implement the above method: need for approximate techniques. At first glance, the above methods can be directly translated into algorithms.

To find out whether two nodes e_1 and e_2 are part of a k-clique, i.e., whether there are $k - 2$ nodes $e_3, \ldots, e_k$ which form a clique, we can try all possible combinations of $k - 2$ nodes. If we denote, by N, the total number of nodes in the graph G, i.e., the total number of entities, then this would require $\binom{N}{k-2} \approx N^{k-2}$ steps.

The problem with this idea is that we are dealing with big data, where the number N of entities is already huge—for example, the US no-fly list containing possible suspects has about a million people in it. For the value $k = 6$ corresponding to intelligence analysis, we will need N^4 computation steps. For $N \approx 10^6$, this leads to $N^4 \approx 10^{24}$ computation steps—way beyond the capabilities of modern computers.

The situation is even worse in the general case, when we look for possible indirect connections. In this case, to check whether the given nodes e_1 and e_2 are connected, a natural idea is to try all possible k-cliques containing e_1, i.e., for all possible tuples of $k - 1$ nodes $e_2, \ldots, e_k$ which, together with the given node e_1, form a k-clique. We need $\binom{N}{k-1} \approx N^{k-1}$ steps, which, for $k = 6$ and $N \approx 10^6$, requires 10^{30} computational steps.

How the above method is algorithmically implemented: idea. First, the papers [8, 9] use the concept lattice algorithms to come up, for each entity e, with a list of the closest ones. Then, for each node e and for each m, we can find a m-*neighborhood* of e—i.e., the set consisting of m closest nodes.

Suppose now that we need to check whether the two nodes e_1 and e_2 are connected by a chain of k-cliques. According to the above method, we need to first find a k-clique containing the node e_1. Since, as we have mentioned, there are too many possible sets of $k - 1$ nodes, instead of looking for all possible nodes, we only look for k-cliques among the m nearest nodes; thus, the value m must be selected in such a way that the resulting amount of possible combinations $\binom{m}{k-1}$ does not exceed the computational ability of the available computer.

In this manner, we find one or more k-cliques containing the node e_1. According to the method, all the nodes in all these k-cliques are thus assumed to be actually directly connected to e_1. One of these nodes should start the next k-clique. How can we select, out of these nodes, the node c_2 which is the most promising to start the new k-clique?

In order to select this node c_2, let us recall that when for some k, we claim that the existence of a k-clique confirms the existence of a true connection, in reality, there is still a probability that the observed "connection" was accidental—this probability is very small but still positive. We then conclude that two nodes related by a chain of k-cliques are actually connected. For this conclusion to be true, *all* the k-cliques must be actually connected. If only one the k-cliques is accidental—the whole conclusion fails. Here, the probability that the conclusion is false is equal to the probability that either the first k-clique is accidental, or that the second k-clique is accidental, etc. The longer the chain, the higher this probability. Thus, it is desirable to construct chains of k-cliques which are as short as possible.

Intuitively, the larger the distance between the two nodes, the longer the chains which connect them. To be more precise, we need to take into account that different links correspond to different distance. What we thus really want to minimize is the overall distance, not just the overall number of steps. If we select a node e' as the nest step c_2, then the overall chain-following distance between e_1 and e_2 can be estimated as the sum of the distance from e to e' and from e' to e_2, i.e., as $D(e_1, e') + D(e', e_2)$. We therefore select a node for which this sum is the smallest possible.

A similar greedy-algorithm idea can be used on the next step, etc. As a result, we arrive at the following algorithm.

How the above method is algorithmically implemented: details. We want to check whether the given nodes e_1 and e_2 are actually connected—and if so, we want to design a chain of events $c_1 = e_1$, c_2,..., c_t, and $c_{t+1} = e_2$ in which each c_i id directly connected to c_{i+1}.

In the algorithm, we start with $c_1 = e_1$, and we select the nodes c_2, c_3,..., c_t one by one. For every i, once the node c_i is selected, we find m nodes which are the closest to c_i. Out of these m nodes, we test all possible subsets of $k - 1$ nodes, and for each subset, we check whether this subset, together with c_i, forms a k-clique. (To be more precise, all m elements have an edge with c_i—otherwise why consider them; thus, it is sufficient to check that the selected $k - 1$ nodes form a $(k - 1)$-clique.) For each subset which leads to a k-clique, we record all its nodes.

- If one of the recorded nodes is e_2, we are done—we have found a chain of k-cliques between e_1 and e_2.
- If none of the recorded nodes coincides with e_2, then out of all recorded nodes e, we select, as the next node c_{i+1} in the chain, the recorded node for which the sum $D(c_i, e) + D(e, e_2)$ is the smallest possible.

If, after a certain number T of steps, we do not teach e_2, we conclude that e_1 and e_2 are not actually connected. (This maximum number of steps T needs to be determined empirically.)

Empirical success. In both applications—to the intelligence analysis and to the biomedical publications—the above method has led to good results, i.e., to the concluded connections for which the high percentage were confirmed by experts as meaningful.

An auxiliary comment: how to gauge our confidence in the results of the method. In general, as we have mentioned, the larger the clique size, the larger our confidence that the nodes are actually connected.

Thus, once we have found that the given nodes e_1 and e_2 are connected by a chain of k-cliques—and thus, we have concluded that e_1 and e_2 are actually connected—we can gauge our degree of confidence in this conclusion by checking whether e_1 and e_2 can be connected by a chain of $(k+1)$-cliques, $(k+2)$-cliques, etc. In this manner, we find the largest click size ℓ for which e_1 and e_2 are connected by a chain of ℓ-cliques. The larger this size ℓ, the more confident we are that e_1 and e_2 are actually connected.

4 Limitations of the Semi-heuristic Approach

First limitation: this method is semi-heuristic. The first limitation is that this method is semi-heuristic: its main justifications are common sense and the fact that in several practical problems, this method was reasonably successful. It is desirable to provide a more formal justification for this method—ideally, a justification which would allow us not only to make conclusions, but also to provide a reasonable estimate of our degree of certainty in this conclusion.

Second limitation: need for flexible granules. The second limitation is that the above semi-heuristic method depends on "crisp" granules (clusters)—namely, k-cliques. As a result:

- If, for some nodes e_1 and e_2, there is a k-clique which contains both e_1 and e_2, then we conclude that e_1 and e_2 are actually directly connected.
- If no such k-clique exists, then we conclude that e_1 and e_2 are not actually directly connected.

From the intuitive viewpoint, this conclusion is too crisp. Intuitively, if we have a subgraphs G which is "almost" a k-clique—i.e., a k-clique with one (or even two) edges missing, it may not affect the conclusion. For example, for $k = 6$, being a k-clique means that we have $\frac{k \cdot k-1}{2} = \frac{6 \cdot 5}{2} = 15$ edges between $k = 6$ nodes; what if we have only 14? There should be a threshold, but this threshold does not necessary mean the threshold between a full k-clique and a graph in which one edge is missing—maybe it is OK if two or more edges are missing?

Right now, the corresponding numerical characteristic—the size k of the largest k-clique connecting two nodes—is too crisp:

- This characteristic decreases rapidly (to $k - 1$) when we delete a single edge from the k-clique.
- And then, when we delete one more edge between some other nodes, this characteristic does not change at all.

It is desirable to generalize a crisp notion of an integer clique size k into a more flexible notion of the fractional-valued "degree" of clique-ness (i.e., the degree of being a granule); see, e.g., [10, 13, 21].

Similarly, for a general connectedness:

- If, for some nodes e_1 and e_2, there is a relating chain of k-cliques, then we conclude that e_1 and e_2 are actually connected.
- If no such chain exists, then we conclude that e_1 and e_2 are not actually connected.

Intuitively, if we have a sequence of subgraphs $G_1, G_2, \ldots$, in which one of the graphs is "almost" a k-clique, it may not affect the conclusion.

The above degree of certainty—the size k of the cliques—is also too crisp:

- If e_1 and e_2 can be related by a chain of k-cliques but cannot be related by a chain of $(k + 1)$-cliques, then our degree of confidence corresponds to k.
- If e_1 and e_2 can be related by a chain of $(k + 1)$-cliques, then our degree of confidence corresponds to the level $k + 1$ (or higher).

What about the situation when we have a chain of graphs $G_1, G_2, \ldots, G_t$ in which all graphs except one are $(k + 1)$-cliques but the remaining one is still a k-clique? According to the above method, we assign, to this case, the degree of certainty k—the same as if all the graphs are k-cliques. However, intuitively, we are almost in the case of $(k + 1)$-cliques, so to this "almost $k + 1$" case, we should be able to assign the degree of confidence which is closer to $k + 1$.

We should also assign different degree of certainty depending on how long is the chain of k-cliques. As we have mentioned, the longer the chain, the less confident we are that this chain implies the actual connection. We used this intuitive idea in designing the algorithm, but this idea is not reflected in how we estimate our degree of confidence—whether we have a chain of length 1 or a chain of the maximally allowed length T, we assign the same degree of confidence k to the conclusion that the corresponding nodes e_1 and e_2 are actually connected. It is desirable to assign the degree of confidence in such a way that longer chains would indeed lead to a smaller degree of confidence.

What we plan to do. We provide an uncertainty-based theoretical statistical framework which enables us, first, to justify the empirical clique approach and, second, to come up with formulas describing to what degree a given subgraph is a granule.

5 Analysis of the Problem and the Resulting Ideas and Formulas

Detecting direct connections based on a graph: analysis of the problem. Let us start with the first part of the problem—detecting direct connections. We will first analyze it in its simplified form—when we ignore the actual distances between the nodes and we only take into account whether the corresponding distance is below the threshold θ or not. In other words, we would like to detect direct connectedness based on a graph G.

As we have mentioned, the fact that there is an edge does not necessarily mean that entities are actually connected; there is a probability r that the edge is accidental. This probability r can be obtained, e.g., by analyzing the part of the graph for which we already know which entities are actually connected and which are not. If in this part of the graph, out of E edges, E_a of them correspond to actual connections, then we can estimate r as the ratio $\frac{E_a}{E}$.

We would like to estimate the probability that the given graph G—in which some entities are linked by an edge and some are not—describes actually connected entities. Let us pick any entity e in this graph. If we already know that all the other entities from G (i.e., the set $G - \{e\}$) are actually connected, then:

- for e to be actually connected to *all* these entities $e' \in G - \{e\}$,
- it is sufficient to show that e is directly connected to *one* of the entities $e' \in G - \{e\}$.

Indeed, if e is actually connected to some $e' \in G - \{e\}$, then, since e' is connected to every other entity from $G - \{e\}$, this would imply that e is actually connected with all the entities from $G - \{e\}$ (and thus, that all the entities from G are indeed connected to each other).

Since at least one actual connection from e to $G - \{e\}$ makes e connected to all other entities from $G - \{e\}$, the only possibility for e to be *not* actually connected to $G - \{e\}$ is when *all* edges between e and elements of $G - \{e\}$ are accidental. In graph theory, the number of edges between a node e and all other nodes is known as the *degree* of a node—and it is denoted by $\deg(e)$. In these terms, e is not connected if all $\deg(e)$ edges are accidental.

The probability that each edge is accidental is equal to r. Since we have no reason to make any conclusion about the dependence between different edges, we will assume that different edges correspond to independent events. If we have two independent or more events, then the probability of them happening together is equal to the product of the corresponding probabilities: e.g., the probability that the coin falls heads three times in a row is the product of the three probabilities corresponding to the three coin tosses, i.e., to $\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8}$. Thus, under the independence assumption, the probability that all deg (e) edges are accidental is equal to the product of $\deg(e)$ probabilities each of which is equal to r—i.e., to $r^{\deg(e)}$.

As a result, the probability that e *is* actually connected to $G - \{e\}$ is equal to $1 - r^{\deg(e)}$.

All the entities from a graph $G = \{e, e', e'', \ldots\}$ are actually connected if each of these entities is connected to all others, i.e., if the entity e is connected to all the other entities, and the entity e' is connected to all the other entities, and the entity e'' is connected to all the other entities, etc.

- We already know the probability that the entity e is actually connected to all other entities from the graph G: this probability is equal to
$$1 - r^{\deg(e)}; \tag{8}$$
- similarly, we know the probability that the entity e' is actually connected to all other entities from the graph G: this probability is equal to
$$1 - r^{\deg(e')}; \tag{9}$$
- we know the probability that the entity e'' is actually connected to all other entities from the graph G: this probability is equal to
$$1 - r^{\deg(e'')}; \tag{10}$$
- and so forth.

It is also reasonable to assume that the corresponding events are independent. Thus, we arrive at the following conclusion.

Detecting direct connections based on a graph: the resulting formula. For each graph G, the probability $P(G)$ that all entities from the graph are actually connected is equal to the product

$$P(G) = \prod_{e \in G} \left(1 - r^{\deg(e)}\right). \tag{11}$$

Alternatively, we can describe the probability $R(G) = 1 - P(G)$ that at least some of the entities from G are *not* connected. This probability is equal to

$$R(G) = 1 - \prod_{e \in G} \left(1 - r^{\deg(e)}\right). \tag{12}$$

As usual in statistical methods, we conclude that all the entities from the graph G are actually connected if this product is greater than or equal to a certain threshold P_0:

$$P(G) = \prod_{e \in G} \left(1 - r^{\deg(e)}\right) \geq P_0. \tag{13}$$

Alternatively, this condition can be described as $R(G) \leq p_0$, where $p_0 \stackrel{\text{def}}{=} 1 - P_0$.

Towards a simplified approximate versions of the formula (13). Usually, the probability r is reasonably small, and for each node e, the number of edges $\deg(e)$ is reasonably large; thus, the probability $r^{\deg(e)}$ is small. In this case, we can expand the expression $\prod_{e \in G}\left(1 - r^{\deg(e)}\right)$ in Taylor series in terms of these small quantities $r^{\deg(e)}$, and keep only linear terms in this expansion.

For two variables, we have

$$(1-a) \cdot (1-b) = 1 - a - b + a \cdot b \approx 1 - (a+b). \tag{14}$$

For three or more variables, we similarly have

$$(1-a) \cdot \ldots \cdot (1-b) \approx 1 - (a + \cdots + b). \tag{15}$$

Thus, we arrive at the following approximate formula.

The resulting simplified approximate versions of the formula (13). For every graph G, the probability $R(G)$ is approximately equal to

$$R(G) \approx \sum_{e \in G} r^{\deg(e)}. \tag{16}$$

Correspondingly, for $P(G) = 1 - R(G)$, we have

$$P(G) \approx 1 - \sum_{e \in G} r^{\deg(e)}. \tag{17}$$

Particular case of a *k*-clique. In the particular case when the graph G is a k-clique, this graph has k nodes for each of which $\deg(e) = k - 1$. In this case, the formulas (13) and (14) takes the form

$$P(G) = \left(1 - r^{k-1}\right)^k; \quad R(G) = 1 - \left(1 - r^{k-1}\right)^k. \tag{18}$$

The simplified approximate formulas (16) and (17) take the form

$$P(G) \approx k \cdot r^{k-1}; \quad R(G) \approx 1 - k \cdot r^{k-1}. \tag{19}$$

Resulting natural definition of a degree of clique-ness. Based on the above formulas (13) and (18), we can define, for each graph, its "degree of clique-ness" as a real number k for which

$$P(G) \overset{\text{def}}{=} \prod_{e \in G} \left(1 - r^{\deg(e)}\right) = \left(1 - r^{k-1}\right)^k. \tag{20}$$

Comment. If we use the simplified approximate expressions for $P(G)$, the above equation for the degree of clique-ness k gets a simplified form:

$$\sum_{e \in G} r^{\deg(e)} = k \cdot r^{k-1}. \tag{21}$$

Example. For $p = 0.1$, for a 6-clique C_6, with $k = 6$, we have $R(C_6) = 6 \times 10^{-5} = 0.00006$. For a 5-clique C_5, we have $R(C_5) = 5 \cdot 10^{-4} = 0.0004$.

If we delete an edge that links two nodes of the 6-clique, then in the resulting graph G, we have two nodes e with $\deg(e) = 4$ and four remaining nodes with $\deg(e) = 5$. Thus, for this graph G, we have $R(G) = 2 \cdot 10^{-4} + 4 \cdot 10^{-5} = 0.00024$.

While this value is larger than the value $R(C_6)$ corresponding to a 6-clique, it is smaller than the value $R(C_5)$ corresponding to a 5-clique: $R(C_6) < R(G) < R(C_5)$. Thus, for the graph G, the above-defined degree of clique-ness is in between 5 and 6—exactly as we wanted it to be.

We thus get a flexible degree of confidence. In contrast to the traditional case, where our degree of confidence was described by a not-very-flexible integer k, now we are allowing non-integer values as well.

- Thus, e.g., if we delete one edge in a large clique, this leads to a minor change in $P(G)$ and thus, to a minor change in k. In contrast, for integers, this was a significant decrease from k to $k - 1$.
- Similarly, if we delete the second edge, we get a new small decrease. In contrast, for integers, we had no change.

If we use the simplified approximate formula, we get an explicit formula for the degree of clique-ness. The above equation for the degree of clique-ness k is similar to the equation that describes Lambert's W-function $W(z)$ (see, e.g., [17]): namely, $W(z)$ is defined as a value w for which $z = w \cdot e^w$.

This formula is similar to the formula that defines k, but it has two differences:

- first, in the formula that defines the W-function, we raise to the power w, while here, we raise r to the power $k - 1$;
- second, in the formula that defines the W-function, we raise e to some power, while here we raise p to some power.

To reduce the above equation to this form, let us transform our formula so as to eliminate these two differences.

First, let us reduce raising to the power $k - 1$ to raising to the power k. For that, we can use the known relation $r^{k-1} = \frac{r^k}{r}$. Substituting this expression into the equation that defines k, we get $R(G) = k \cdot \frac{r^k}{r}$, or, equivalently, $k \cdot r^k = r \cdot R(G)$.

To reduce raising r to some power to raising e to some point, we take into account that, by definition of the natural logarithm, the value r can be described as $e^{\ln(r)}$. Thus, $r^k = \left(e^{\ln(r)}\right)^k = e^{k \cdot \ln(r)}$. Hence, our equation takes the form $k \cdot e^{k \cdot \ln(r)} = R(G) \cdot r$. Here, e is raised to the power $w \stackrel{\text{def}}{=} k \cdot \ln(r)$, i.e., we have $r^k = e^w$. We can explicitly describe k in terms of w, as $k = \frac{w}{\ln(r)}$. Substituting the above expressions for r^k and k in terms of w into the equation $k \cdot r^k = r \cdot R(G)$, we conclude that $\frac{w}{\ln(r)} \cdot e^w = R(G) \cdot r$, i.e., that $w \cdot e^w = R(G) \cdot r \cdot \ln(r)$. Thus, by definition of the W-function, we have $w = W(R(G) \cdot r \cdot \ln(r))$, and hence, for the desired degree of clique-ness $k = \frac{w}{\ln(r)}$, we get an explicit formula

$$k = \frac{1}{\ln(r)} \cdot W(R(G) \cdot r \cdot \ln(r)). \tag{22}$$

What if we have a chain of subgraphs? In general, we have a *chain* of graphs $G_1, \ldots, G_t$ linking two entities e_1 and e_2. To be able to conclude that e_1 and e_2 are actually connected, we need to be able to conclude:

- that the first graph G_1 corresponds to the actual connection,
- that the second graph G_2 corresponds to the actual connection,
- etc.

For each graph G_i, we have already estimated the probability $P(G_i)$ that this graph corresponds to actual connections. Similarly to the above situations, it is reasonable to assume that the corresponding events are independent. Thus, the probability C that e_1 and e_2 are actually connected—i.e., the probability that all the graphs in the chain correspond to actual connections—can be estimated as the product of the corresponding probabilities:

$$C = \prod_{i=1}^{t} P(G_i). \tag{23}$$

Comment. In particular, if we take into account that $P(G_i) = 1 - R(G_i)$ and that the values $R(G_i)$ are small, we can use a similar approximation as above and get an approximate formula

$$C \approx 1 - \sum_{i=1}^{t} R(G_i). \tag{24}$$

This enables us to gauge how our confidence that e_1 and e_2 are connected decreases when the chain gets longer. In the formula (23), our degree of confidence that e_1 and e_2 are connected is equal to the product of the probabilities $P(G_i)$ corresponding to all the graphs G_i in the chain relating e_1 and e_2. Each multiplication by the number $P(G_i) < 1$ decreases the product. The longer the chain, the smaller the product and thus, the smaller our degree of confidence that e_1 and e_2 are actually connected.

This solves one of the problems that we mentioned—that, contrary to intuition, in the semi-heuristic approach, the degree of confidence (as described by the clique size) does not decrease when the length of the chain increases.

6 Towards an Algorithm

How to take distance into account when estimating the probability: idea. As we have described earlier, the existing algorithm for checking when the two nodes are actually connected uses the distances, not just the graph. We therefore need to extend the above probabilistic analysis so that it takes into account the actual distances, not just whether there is an edge or not.

In the graph version, we assumed that there is a probability r that the edge between the nodes is accidental—and does not reflect the true connection between the nodes. Since an edge is placed when the distance is $\leq \theta$, we thus assign the probability r to all distances $D \leq \theta$—and this value immediately jumps to 1 when the distance exceeds θ and therefore, there is no edge. The true probability should not change that abruptly, especially since the value θ has to be empirically determined—and may thus change from situation to situation.

In other words, instead of a *single* probability value r, we should come up with the value $r(D)$ *depending on the distance*—and make sure that this dependence on D is continuous, with no abrupt jumps. This function should be non-decreasing:

- when the distance increases,
- the probability that the entities are not actually connected should also increase (or at least not decrease),

i.e., $D \leq D'$ should imply $r(D) \leq r(D')$.

To find such a function, let us consider the situation in which a node e' is in between nodes e and e'', in the sense that $D(e, e'^{e}) = D(e, e') + D(e', e'')$, i.e., the distance $D(e, e'')$ is equal to the sum $D + D'$, where we denoted $D \overset{\text{def}}{=} D(e, e')$ and $D' \overset{\text{def}}{=} D(e', e'')$. By definition of the function $r(D)$:

- the probability that the entities e and e' are actually connected is equal to $1 - r(D)$;
- the probability that the entities e' and e'' are actually connected is equal to $1 - r(D')$; and

- the probability that the entities e and e'' are actually connected is equal to $1 - r(D + D')$.

The nodes e and e'' are actually connected if both e is connected to e' and e' is connected to e''. Similar to the previous parts of this chapter, it is reasonable to assume that the corresponding events are independent. Thus, we get

$$1 - r(D + D') = (1 - r(D)) \cdot (1 - r(D')). \tag{25}$$

Thus, a non-increasing function $p(D) \overset{\text{def}}{=} 1 - r(D)$ satisfies the functional equation $p(D + D') = p(D) \cdot p(D')$.

It is known (see, e.g., [1]) that all the solutions of such an equation have the form $p(D) = \exp(-a \cdot D)$ for some constant $a > 0$. Thus, we arrive at the following conclusion.

How probability depends on the distance. The probability $p(D)$ that two nodes are actually connected is equal to $p(D) = \exp(-a \cdot D)$ for some constant $a > 0$.

The parameter a needs to be determined empirically, based on the part of our data for which we already know which entities are actually connected and which are not.

The probability $r(D) = 1 - p(D)$ that there is no connection between the two nodes is therefore equal to $r(D) = 1 - \exp(-a \cdot D)$.

Detecting direct connections: case when we take distances into account. Similar to the graph case, we first compute, for each node e, the probability that all connections from e to nodes from $G - \{e\}$ are accidental. Just like in the graph case, this probability is equal to the product of the probabilities $\exp(-a \cdot D(e, e'))$ that the distance between e and e' does not imply an actual connection. This product is equal to $\prod_{e' \neq e} \exp(a \cdot D(e, e'))$.

This formula can be simplified.

- First, we can easily add $e' = e$ to the product, since for $e' = e$, we have $D(e, e') = 0$ and thus, the factor $\exp(-a \cdot D(e, e)) = 1$ does not change the overall product.
- Second, we can use the fact that the product of the exponents is equal to the exponent of the sum. As a result, we get a simplified formula

$$\exp\left(-a \cdot \sum_{e' \in G} D(e, e')\right). \tag{26}$$

Thus, the probability that e *is* actually connected to $G - \{e\}$ is equal to

$$1 - \exp\left(-a \cdot \sum_{e' \in G} D(e, e')\right). \tag{27}$$

The probability $P(G)$ that all nodes from G are actually connected can be now estimated as the product of the probabilities corresponding to different nodes $e \in G$:

$$P(G) = \prod_{e \in G}\left(1 - \exp\left(-a \cdot \sum_{e' \in G} D(e, e')\right)\right). \tag{28}$$

Comment. In the first approximation, we get a simplified formula

$$P(G) \approx 1 - \sum_{e \in G} \exp\left(-a \cdot \sum_{e' \in G} D(e, e')\right). \tag{29}$$

Towards an algorithm. We start building a chain with $c_1 = e_1$. In the original method, we only considered k-cliques; now, we are allowing graphs which are "almost" cliques.

For each such graph, we can use the formula (29) to estimate the probability $P(G)$ that this nodes from this graph are actually connected. For each node e' from this graph, we probability that it is actually connected to c_1 is equal to $p(G)$ and the probability that it is actually connected to e_2 is equal to

$$\exp(-a \cdot D(e', e_2)). \tag{30}$$

Thus, the probability that e_1 and e_2 are connected via e' is equal to the product of these two probabilities, i.e., to $P(G) \cdot (1 - \exp(-a \cdot D(e', e_2)))$. As the next node in the connecting chain, we then select the most probable connecting node e', i.e., the node for which this product is the largest possible.

Then, we repeat the same procedure starting with c_2, etc., until we reach e_2. As a result, we arrive at the following algorithm.

7 Resulting Algorithm

Formulation of the problem: reminder. We want to check whether the given nodes e_1 and e_2 are actually connected—and if yes, we want to design a chain of events $c_1 = e_1$, c_2,…, c_t, and $c_{t+1} = e_2$ in which each c_i is directly connected to c_{i+1} (and the corresponding chain of connecting graphs G_1,…,G_t).

We also want to compute the probability P that the corresponding chain reflects the actual connection.

First preliminary step: finding the parameter $a > 0$. Based on the part of the data for which we already know which entities are actually connected and which are not, we estimate the parameter $a > 0$ for which the probability $p(D)$ that nodes at distance D are actually connected decreases as exp $(-a \cdot D)$.

This value can be estimated, e.g., if for different values d, we estimate, among all pairs nodes of distance approximately D, the proportion $\widetilde{p}(D)$ of pairs were actually connected. Then, we try to find a for which, for all these values D, we have $\widetilde{p}(D) \approx \exp(-a \cdot D)$. To estimate a, we can, e.g., take negative logarithm of both sides, and use the Least Squares Method (see, e.g., [18]) to solve the resulting system of approximate linear equations $a \cdot D \approx -\ln(\widetilde{p}(D))$.

Second preliminary step: finding neighborhoods. Similar to [8, 9], use the concept lattice algorithms to come up, for each entity e, with a list of the closest ones. Then, for each node e and for each m, we can find a *m-neighborhood* of e—i.e., the set consisting of m closest nodes. For this, we can use, e.g., an algorithm for computing the concept lattice (as in [8, 9]).

The corresponding value m and the value k (which is used in the main part of the algorithm) are chosen in such a way that it is computationally feasible to try all possible subsets of $\leq k - 1$ elements out of m.

Main part of the algorithm. We start with $c_1 = e_1$. Then, we select the nodes $c_2, c_3, \ldots, c_t$ one by one.

When we reach the node c_i, we estimate the probability P_i that c_1 and c_i are actually connected. We start with the probability $P_1 = 1$ (reflecting the fact that the node e_1 is clearly connected to itself).

For every i, once the node c_i has been selected and the value P_i has been computed, we find m nodes which are the closest to c_i. Out of these m nodes, we test all possible subsets of $\leq k - 1$ nodes. To each of these subsets, we add the node c_i and consider the corresponding graph G. For this graph G, we compute the probability

$$P(G) = \prod_{e \in G} \left(1 - \exp\left(-a \cdot \sum_{e' \in G} D(e, e') \right) \right). \tag{31}$$

Then, for each point $e' \in G - \{c_i\}$, we compute the product

$$P(G) \cdot (1 - \exp(-a \cdot D(e', e_2))). \tag{32}$$

Once we have tested all such subsets G and computed the product for all their elements $e' \in G$, we select, as the next node c_{i+1} in the chain, the node e' for which the product corresponding to this node is the largest possible. The corresponding graph G is selected as the connecting graph G_i. We then compute $P_{i+1} = P_i \cdot P(G_i)$.

- If the probability P_{i+1} goes below a certain threshold P_0, we conclude that e_1 and e_2 are not actually connected (or, to be more precise, that, based on the available information, we cannot make such a conclusion).

- If $c_{i+1} = e_2$ and $P_{i+1} \geq P_0$, then we conclude that the given nodes e_1 and e_2 are actually connected, with degree of confidence $P = P_{i+1}$.
- If $c_{i+1} \neq e_2$ and $P_{i+1} \geq P_0$, we continue iterations.

Experimental results. As we have mentioned earlier, this algorithm has led to successful discovery of connections in intelligence analysis [8] and in the analysis of biomedical publications [9]. In both cases, the algorithm, by using only the information about joint appearance in documents, was able to uncover important relations between the corresponding objects. The fact that in these two examples, we were able to uncover previously known useful relations makes us believe that this technique will enable other users to uncover relations of importance.

8 Conclusions

In many practical situations, it is important to check which entities are actually connected and which are not. Usually, this checking is performed by using the traditional statistical methods—but these methods cannot be applied when we have a large amount of data points ("big data"). A semi-heuristic method was proposed to detect actual connections in the case of big data; however this method has limitations: first, it is justified by experimental results and requires theoretical justification, and second, the method depends on "crisp" granules (cliques) to form connections.

In this chapter, we have come up with a theoretical justification of the known semi-heuristic method, and we have come up with a new, more flexible definition of almost-granules. However, a lot of work is still ahead: there is still a lot of room for improvement in how we can effectively process big data to find such almost-granules and to compute their degree of granule-ness.

Acknowledgements This work was supported in part by the National Science Foundation (NSF) grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence), NSF grant DUE-0926721, and by M. S. Hossain's startup grant at UTEP. The authors are greatly thankful to the anonymous referees for valuable suggestions and to the editors of this volume, Shyi-Ming Chen and Witold Pedrycz, for their support and encouragement.

References

1. Aczel, J.: Functional Equations and Their Applications. Academic, New York (1966)
2. Brassard, J.-P., Gecsei, J.: Path building in cellular partitioning networks. ACM SIGARCH Computer Archit News **8**(3), 44–50 (1980)
3. Di Ciaccio, A., Coli, M., Angulo Ibanez, J.M. (eds.): Advanced Statistical Methods for the Analysis of Large Data. Springer, Berlin (2012)

4. Faloutsos, C., McCurley, K.S., Tomkins, A.: Fast discovery of connection subgraphs. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD'04, Seattle, Washington, pp. 118–127. 22–25 Aug 2004
5. Fang, L., Sarma, A.D., Yu, C., Bohannon, P.: Rex: explaining relationships between entity pairs. Proc. VLDB Endowment **5**(3), 241–252 (2011)
6. Heath, K., Gelfand, N., Ovsjanikov, M., Aanjaneya, M., Guibas, L.: Image webs: computing and exploiting connectivity in image collections. In: Proceedings of the 23th IEEE Conference on Computer Vision and Pattern Recognition CVPR'2010, San Francisco, California, pp. 3432–3439. 13–18 June 2010
7. Hossain, M.S., Akbar, M., Polys, N.F.: Narratives in the network: interactive methods for mining cell signaling networks. J. Comput. Biol. **19**(9), 1043–1059 (2012)
8. Hossain, M.S., Butler, P., Boedihardjo, A.P., Ramakrishnan, N.: Storytelling in entity networks to support intelligence analysts. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD'12, Beijing, China, pp. 1375–1383. 12–16 Aug 2012
9. Hossain, M.S., Gresock, J., Edmonds, Y., Helm, R., Potts, M., Ramakrishnan, N.: Connecting the dots between PubMed abstracts. PLoS ONE **7**(1), Paper e29509 (2012)
10. Klir, G., Yuan, B.: Fuzzy Sets and Fuzzy Logic. Prentice Hall, Upper Saddle River (1995)
11. Kumar, D., Ramakrishnan, N., Helm, R., Potts, M.: Algorithms for storytelling. IEEE Trans. Knowl. Data Eng. **20**(6), 736–751 (2008)
12. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
13. Nguyen, H.T., Walker, E.A.: A First Course in Fuzzy Logic. Chapman and Hall/CRC, Boca Raton, Florida (2006)
14. Ohlhorst, F.J.: Big Data Analytics. Wiley, New York (2012)
15. Pedrycz, W.: Granular Computing: Analysis and Design of Intelligent Systems. CRC Press/Francis Taylor, Boca Raton (2013)
16. Rajaraman, A., Ullman, J.D.: Mining of Massive Datasets. Cambridge University Press, Cambridge (2011)
17. Roy, R., Olver, D.W.J.: Lambert W function. In: Olver, W.J., Lozier, D.M., Boisvert, R.F., Clark, C.F. (eds.) NIST Handbook of Mathematical Functions. Cambridge University Press, Cambridge (2010)
18. Sheskin, D.J.: Handbook of Parametric and Nonparametric Statistical Procedures. Chapman and Hall/CRC Press, Boca Raton, Florida (2011)
19. Srinivasa, S., Bhatnagar, V. (eds.): Big data analytics. In: Proceedings of the First International Conference on Big Data Analytics BDA'2012. Lecture Notes in Computer Science, vol. 7678. Springer, New Delhi, 24–26 Dec 2012
20. Swanson, D.R.: Complementary structures in disjoint science literatures. In: Bookstein, A., Chiaramella, Y., Salton, G., Raghavan, V.V. (eds.) Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR'91, Chicago, Illinois, pp. 280–289. 13–16 Oct 1991
21. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**(3), 338–353 (1965)

Graph-Based Framework for Evaluating the Feasibility of Transition to Maintainomics

Bo Xing

Abstract Maintenance is a powerful support function for ensuring equipment productivity, availability and safety. Nowadays, growing concern for timeliness, accuracy and the ability to offer tracking information led to the augmentation of e-technologies' applications within maintenance management, i.e., e-maintenance. However, like any other information and communication (ICT)-based operation, massive data sets (i.e., big data) are generated from videos, audios, images, search queries, historic records, sensors, etc. Inevitably, e-maintenance needs to consider how to extract useful value from those raw and/or fused data as an important aspect before it can be adopted in any industry. This book chapter presents an overview of the e-maintenance data challenge. The main contribution of the article is the application of graph-theoretic approach (GTA) to the problem of finding an improved insight in the factors that determine the feasibility of maintainomics, i.e., data-centric maintenance. With such a concept, the maintenance-services can be upgraded from the low level of operations to the higher levels of planning and decision making.

Keywords e-Maintenance · Maintainomics · Graph-theoretic approach (GTA) · Feasibility index of transition (FIT) · Power plant · Innovative computational intelligence

B. Xing (✉)
Center for Asset Integrity Management (C-AIM), Department of Mechanical and Aeronautical Engineering, Faculty of Engineering, Built Environment and Information Technology, University of Pretoria, Pretoria, South Africa
e-mail: bxing2009@gmail.com

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_5,

1 Introduction

From a country's perspective, critical infrastructures may include such as banks, railways, and power supplies. Each of these infrastructures is so "large" and therefore can also be called as system of system since any one of them is a collection of task-oriented or dedicated sub-systems. All these sub-systems are organized in a manner that their resources and capabilities are pooled together to form a new and more complicated main system. Taking the electricity generation industry as an example, on one hand, the healthy condition of the national grid play a key part in keeping a country functioning properly; on the other hand, every involved power plant (more formally, sub-system) is composed of a large amount of equipments in which each individual machine acts an important role in helping the company to retain competitive in the market. As we can see, no matter how "big" or how "small" (relatively of course) a system is, maintaining its functional status is a maintenance engineer's major task. Looking at the power plant case again, traditionally, there are a number of ways in which power plant assets (e.g., transformers, turbines, and pipelines) fail and eventually get replaced. "Fail and fix" is often a practiced strategy to cope with this situation. However, as the competition in the global market increases, the accumulated breakdowns occurred in one single power plant may finally generate a fatal impact on the whole power supply system's performance. Under these circumstances, there is a need to consider all aspects of a sub-system performance, such as improving equipment reliability and availability, reducing unplanned outages, and predicting the remaining life of key system components. In this regard, a critical issue turns into selecting proper information processing and communicating tools to support the inspection and maintenance management. That means, for any an organization, maintenance interventions have to migrate from their traditional reactive approach, namely, "fail and fix", to a more advanced proactive approach, i.e., "predict and prevent" [1].

One possible direction which may spawn new ways to manage maintenance activities is the adoption and the usage of pervasive digital communications, e.g., mobile devices, remote sensing, online condition monitoring, etc. Accordingly, a new maintenance strategy (i.e., maintainomics) can be envisaged in which maintenance tasks are managed electronically by analyzing various real-time data [2]. For instance, Ref. [3] presented an integrated approach towards e-maintenance of engineering assets that based on radio frequency identification (RFID) technology. In a similar vein, the authors of [4] focused on the research of intelligent maintenance decision-making tools (i.e., Watchdog Agent$^{\text{TM}}$) which is used for multi-sensor assessment and prediction of a machine's or process's performance. Although maintainomics is desirable from many perspectives, all those digital bits that have been gathered are at the same time possible undesired and hazardous, just because more data (i.e., big data) does not necessarily mean better insights, in some cases, it even may result in confusion or disaster. Consequently, it is important to think strategically about how to adapt maintainomics to meet new maintenance demands. To address this issue, a graph-theoretic approach (GTA) is used to identify dominant factors that determining the feasibility of such transformation in the era of big data.

The remainder of this chapter is organized as follows. Subsequent to the introduction in Sect. 1, the background of maintenance, e-maintenance, big data, and maintainomics are briefed in Sect. 2. Then, the problem statement is presented in Sect. 3. The proposed methodologies are then detailed in Sect. 4. Next, Sects. 5 and 6 conduct an experimental study to demonstrate the feasibility of our proposed approaches. The future research directions are highlighted in Sect. 7. Finally, the conclusions drawn in Sect. 8 close this chapter.

2 Background

2.1 What is Maintenance?

Maintenance is normally seen as one of the few opportunities to reduce the cost of production, because it is the second highest operation costs in some industries [5]. Briefly, the heart of maintenance processes is condition monitoring that includes data acquisition, processing, analysis, interpretation, and extracting useful information from it. Traditionally, the maintenance actions are performed only when there is evidence of abnormal behaviors of a physical asset [6]. At the same time, researchers and/or scientists have relied upon monitoring programs just using invasive sampling at discrete periods, good experts, and/or handbooks. Limitations associated such programs (e.g., good experts are rare, sample acquisition takes long time, and uncertainty embedded for tedious sample analysis) result in high risk of equipment failures and therefore the company's top-line revenue plummets. In addition, the previous condition monitoring focuses only on monitoring and diagnostics, ignore prediction and prognosis [7]. As a result, to perform predictive-prognostic maintenance, it is evident that a proactive as well as reactive e-maintenance support system is required.

2.2 What is e-Maintenance?

Nowadays, the dependence on remotely sensed information can be found in many crucial areas of human endeavor such as meteorology, security services, banking systems, supply chains, and scientific researches. In asset management area, several authors pointed out that information and communication technologies (ICT) can be adopted as a tool to support in terms of quality data acquisition, real-time monitoring, and recording of divergences from standard acquisition [3]. In the literature (e.g., [5, 8–13]), this concept is defined as e-maintenance or tele-maintenance.

For instance, a new e-maintenance system has been studied in [14] which is based on the use of Internet and tether-free (i.e., wireless, Web, etc.) communication technologies. Later, Ref. [15] focused on the implementation of Web-based techniques to support e-maintenance services. In a similar vein, the authors of [16] proposed a framework by using RFID technology for real-time management of mobile assets. Also, worldwide case studies (e.g., automobile industry [17], power plant [18]) convinced that the implementation of ICT tools can improve products quality and reduce annual costs for maintenance.

In addition, as the maintenance itself is an extremely complex process and sometimes the inspections are difficult carried out by human operators even performed by dexterous technicians, the using of remotely controlled intelligent robots in an effort to accomplish the inspection or maintenance jobs faster and more reliable has increased greatly in the last few decades. For example, the authors of [19, 20] developed climbing robot for the structural inspection. An intelligent legged climbing robot is designed by [21] to perform the required inspection work in hazardous environments. Meanwhile, a large amount of pipeline inspection robots (i.e., in-pipe robots) have also been designed and fabricated. Generally speaking, the in-pipe robots can be classified into caterpillar [22], inchworm [23], walking [24], wheel [25], and pig types [26] depending on their travelling mechanisms. Interested readers please refer to [27, 28] for more detailed information regarding the intelligent robot assisted e-maintenance.

2.3 What is Big Data?

Recent advances in intelligence products, such as smart phones, and sensors and tracking devices, create tremendous amount of data (i.e., big data), which enable a researcher to extract the meaningful value much easier. According to a recent report compiled by McKinsey, the term “big data” refers to “datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyses [29] ”. For example, in genomic research, there are approximately 3 billion base pairs with a personal genome representing approximately 100 gigabytes of data [30]. In e-business, the datasets include social networks, purchase transaction records, blogs, mobile telephony, and digital entertainment. Also, from the maintenance engineers’ point of view, the large historical records, the regular collected digital images, and the transactional information for operational reporting are also comprised of huge data. Other important aspect has been emphasized in addition to the volume is that big data may be unstructured, examples are text with social sentiments, audio and video, click streams, and website log files [31]. At the same time, big data is characterized by velocity, i.e., the rate of generation of data. For instance, to determine real-time roadway traffic conditions. As a result, the era of big data needs more interdisciplinary and multi-perspective research approaches that researchers have to create which were hard to implement before, such as new tools and solution approaches for data analytics [32], new paradigm for big data collection, storage, and processing [33], and privacy and security issues with big data [34].

2.4 What is Maintainomics?

As the maintenance environment more and more rely on the mobile devices and sensors, a serious question raises, namely, a huge amount of maintenance related data (big data in nature and is normally obtained by different kinds of automated sensors) are flooding in at rates never seen before. For instance, data from various control sensors for maintaining the different functions. Under these data overloaded circumstances, different sources of data are collected, although a pretty good news compared with the traditional data scarce maintenance scenario, it is still necessary to sort out the required and useful data. In fact, the data in this type of "wide-area" sensing applications are no longer just focus on the equipment itself but can involve the analysis at scale of environment, legislation, and even workers' safety. In this study, the term "maintainomics" is utilized to denote the use of very large maintenance data content and the application of advanced core techniques (e.g., data mining, nature-inspired computational algorithms, Web and mobile technologies, new sensors, wireless communications, and different decision supporting systems) to improve the ways of organizations operating its maintenance management. For example, Ref. [35] proposed a conceptual model of a generic data acquisition system for improving maintenance management. Reference [36] broadly suggested that a common database can play an important role in reaching cost-effective improvements of maintenance performance, while Ref. [37] focused on the issue of maintenance-related data integration. Briefly, in the era of big data, maintainomics is an integration of traditional condition based maintenance, modern real-time online monitoring, and advanced intelligent technologies (see Fig. 1 for illustration). Notations ① ~ ⑥ represent different sources of data collected during the process of e-maintenance which form a more complex scenario, i.e., maintainomics.

3 Problem Statement

Confronting with the emerging e-maintenance and the forthcoming maintainomics operating environment, it is necessary for every organization's strategic planners to make a decision about whether to perform such transformation for their own institution. Nevertheless, reaching a proper decision is often a complex process in practice, for instance, several authors (e.g., [38, 39]) believed in employing e-maintenance concept is not only based on the possibilities that new ICTs could offer but also need to integrate business performance such as operational, financial,

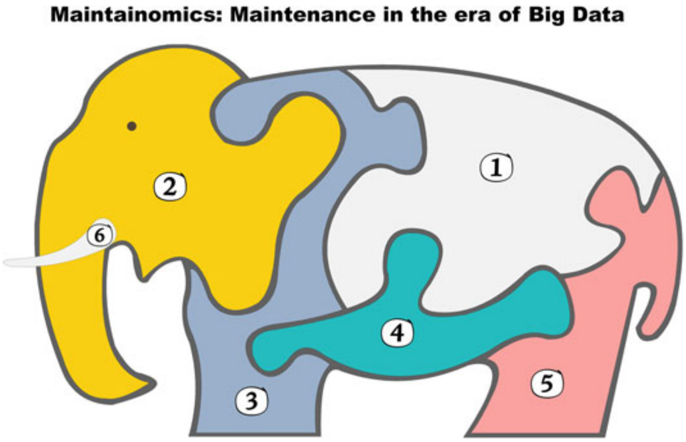


Fig. 1 Maintainomics: maintenance in the era of big data

human, and cultural factors. Consequently, a question raised by this study is how a company's "fitness" can be measured in terms of implementing maintainomics strategy. In other words, we need to come up a solution about how to perform a feasibility assessment for a particular firm regarding its suitability of aligning its present asset management policy with the maintainomics goal.

4 Proposed Methodology

Nowadays, a number of approaches have been suggested in the literature for the purpose of feasibility analysis. Amongst them, the graph theoretic approach (GTA) is often widely used to cope with multi-criteria decision making problems in both academic research and in industrial practice [40–43]. Inspired by the studies conducted by these scholars, the author of this work makes an attempt to apply a recently proposed GTA based methodology (known as feasibility index of transition, or FIT for short) to a selected industry in exploring its feasibility index during the course of transforming into maintainomics, i.e., data enriched e-maintenance. The following subsections give readers a brief overview regarding FIT and its main theoretical foundation, i.e., graph theory.

4.1 Background of Graph Theory

Graph theoretic approach (GTA) is a powerful decision making tool used to represent the relationships among different variables or subsystems based on the form of a digraph (directional graph) and matrix. Since 1736, GTA has been applied in a variety of fields, such as conceptual modelling [44], diagnosis [45], functional representation [46], and network analysis [47]. Basically, all graphs are composed exclusively of vertices (e.g., nodes, points) and edges (e.g., arcs, connections). To allow mathematical analysis, the interdependence on each other as well as their individual contribution to the system is assigned numerical values and an overall index is calculated. Interested readers are referred to [48–50] for more information regarding GTA.

4.2 Background of Feasibility Index of Transition

The feasibility index of transition (abbreviated as FIT) was coined by [51] for describing a chosen company's feasibility of moving forward to flexible manufacturing system (FMS). In order to analyze the trade-offs between various organizations for the adoption of FMS, certain enablers (or resources) should be chosen. In addition, the heritability of available enablers and the quantity of interactions among them may be directional independent or dependent. Built on this concept, a graph representation, illustrating the involved enablers and their potential interaction, was proposed in [51]. If interactions are found to be non-directional dependent, an undirected graph is thus used; on the opposite case, a digraph depiction is then employed.

For example, in their work [51], six groups of enablers (i.e., behavior enablers, non-behavioral enablers, financial enablers, methodologies, operational enablers, and human and cultural enablers) and their corresponding components were first identified and later used to examine the likelihood of transforming a traditional manufacturing environment into more advanced FMS. By simply calculating an index (i.e., FIT) value, one can easily find out to what degree a chosen target "fits" a company's situation. Mathematically, this process can be expressed via Eq. (1) [51]:

$$\text{FIT for ``FMS''} = f(\text{Enablers}) \tag{1}$$

where "FMS" stands for the goal that a manufacturing company wants to achieve.

A more general form of such equation can thus be further written as Eq. (2):

$$\text{FIT for ``Targeted Scenario''} = f(\text{Enablers}) \tag{2}$$

where "Targeted Scenario" represents a generic situation that an organization plans to orient itself to.

To check an organization's "fitness" in transformation, the graph theoretical based approach (utilized in [51]) is composed of four steps, namely, digraph visualization, matrix expression, permanent function establishing, and FIT value scale creating. The following subsections will provide a description of each step's detailed task.

4.2.1 Visualizing Enablers' Correlation via Digraph

In order to visualize the correlation between different enablers, a digraph (i.e., directed graph) is introduced at this stage as shown in Eq. (3) (adapted from [51]):

$$\begin{aligned} G &= (E, e) \\ E &= \{E_1, E_2, \ldots, E_P\} \\ e &= \{e_1, e_2, \ldots, e_P\} \end{aligned} \tag{3}$$

where G refers to a digraph which consists of two sets of items, namely, vertices (denoted by E), and edges (represented by e). In general, each edge (i.e., e_{ij}) can be identified via an ordered pair of vertices, (i.e., (E_i, E_j). The vertices (E_i and E_j) associated with the corresponding edge (e_{ij}) are thus called the end vertices or the nodes of e_{ij}. Please note, a self-loop occurs when a particular edge has the self-same vertex acting as its both nodes.

In [51], a node (E_i) stands for the ith enabler and the interdependence between enablers is pictured by different connecting directed edges. The number of nodes [i.e., the subscript P in Eq. (3)] is equal to the total number of enabler categories. In other words, P is a scenario-dependent parameter which was set to 6 (matching the total enabler categories' amount) in [51]. Accordingly, if a node E_i impresses a certain degree of influence against another node E_j, a directed line (represented by an arrow →) is drawn (from node E_i to E_j) to form edge e_{ij}. An illustration of this visualization process can be found below where an FMS enabler digraph (consisting of six enabler categories) is depicted [51].

As shown in Fig. 2, directional edges are found existing between E_1 and E_2, E_3, E_4, E_5, and E_6, respectively, which means that all other five enabler categories are swayed by node E_1 (i.e., behavioral enabler category) to some extent. For other edges, the same rule applies as well.

As one can see, this quickly drawn digraph can assist the decision makers in visualizing and analyzing the proposed transformation plan from a holistic perspective. Nevertheless, with the ever increasing number of nodes and their associated interrelationship degree, the digraph becomes more and more complex which will in turn decrease its readability. To cope with issue, a matrix form expression is further discussed as below.

4.2.2 Interpreting Enablers' Digraph Through Matrix

A matrix is a handy and powerful way of representing a digraph that is more processable by a computer. Suppose there is a digraph with P enablers, a matrix $\mathbf{F} = \left[e_{ij}\right]$ is thus can be used to represent such digraph as expressed in Eq. (4) [51]:

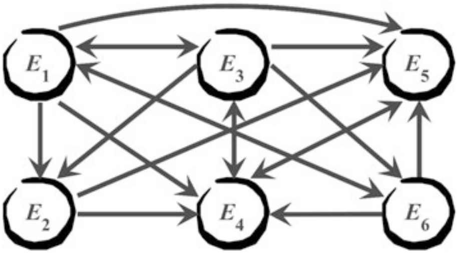


Fig. 2 FMS enabler categories' digraph

$$\mathbf{F}=\begin{array}{cc} \begin{array}{c}\text{Enabler}\\ \text{Categories}\end{array} & \begin{array}{cccccc} E_1 & E_2 & E_3 & \cdots & \cdots & E_P\end{array} \\ \begin{array}{c} E_1\\ E_2\\ E_3\\ \vdots\\ \vdots\\ E_P\end{array} & \begin{pmatrix} E_1 & e_{12} & e_{13} & \cdots & \cdots & e_{1P} \\ e_{21} & E_2 & e_{23} & \cdots & \cdots & e_{2P} \\ e_{31} & e_{32} & E_3 & \cdots & \cdots & e_{3P} \\ \vdots & \vdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ e_{P1} & e_{P2} & e_{P3} & \cdots & \cdots & E_P \end{pmatrix}\end{array} \tag{4}$$

where e_{ij} indicates the interaction between the ith and jth enablers. Built on these principles, an demonstrating matrix expression derived from [51] is shown in Eq. (5):

$$\mathbf{F}^{*}=\begin{array}{cc} \begin{array}{c}\text{Enabler}\\ \text{Categories}\end{array} & \begin{array}{cccccc} E_1 & E_2 & E_3 & E_4 & E_5 & E_6\end{array} \\ \begin{array}{c} E_1\\ E_2\\ E_3\\ E_4\\ E_5\\ E_6\end{array} & \begin{pmatrix} E_1 & e_{12} & e_{13} & e_{14} & e_{15} & e_{16} \\ 0 & E_2 & 0 & e_{24} & e_{25} & 0 \\ e_{31} & e_{32} & E_3 & e_{34} & e_{35} & e_{36} \\ 0 & 0 & e_{43} & E_4 & e_{45} & 0 \\ 0 & 0 & 0 & e_{54} & E_5 & 0 \\ e_{61} & 0 & 0 & e_{64} & e_{65} & E_6 \end{pmatrix}\end{array} \tag{5}$$

where the contributions of six enabler categories in transforming the traditional manufacturing system into FMS are represented via the diagonal elements of the matrix $\mathbf{F}^{*}$, i.e., E_1, E_2, E_3, E_4, E_5, and E_6, respectively. The interdependence of enablers E_i and E_j is denoted by the connecting edge e_{ij}. Please note that $e_{ij} \neq e_{ji}$ since the enablers are directed, and $e_{ii} = 0$ because there is no self-loop existing (in other words, the interaction between an enabler and itself does not exist).

Although the understandability of a matrix expression is better than digraph's from the machine perspective, both representations' uniqueness is inherent low which means they suffer high alterability by simply modifying the labels of their nodes. Under these circumstances, a subsequent fix solution is to establish a permanent function as discussed in subsection below.

4.2.3 Establishing the Matrix's Permanent Function Expression

For the purpose of developing a unique representation that is independent of labeling, a permanent function based on previously established variable matrix was proposed by the author of [51] at this stage. As mentioned in [51], the permanent function (i.e., $f^{\mathbf{F}^*}_{permanent}$) is a standard matrix function and plays a starring role in combinatorial mathematics.

Based on this concept, the permanent function of matrix $\mathbf{F}^*$ can be written as follows (adapted from [51]):

$$f^{\mathbf{F}^*}_{permanent} = item_1 + item_2 + item_3 + item_4 + item_5 + item_6 + item_7$$

$$\text{where}\begin{cases}
item_1 = \prod_{i=1}^{6} E_i \\
item_2 = \prod_{i=1}^{6} e_{ii} \\
item_3 = \sum_{i,j,k,l,m,n} \left(e_{ij}e_{ji}\right) E_k E_l E_m E_n \\
item_4 = \sum_{i,j,k,l,m,n} \left(e_{ij}e_{jk}e_{ki} + e_{ik}e_{kj}e_{ji}\right) E_l E_m E_n \\
item_5 = \sum_{i,j,k,l,m,n} \left(e_{ij}e_{ji}\right)\left(e_{kl}e_{lk}\right) E_m E_n + \\
\qquad \sum_{i,j,k,l,m,n} \left(e_{ij}e_{jk}e_{kl}e_{li} + e_{il}e_{lk}e_{kj}e_{ji}\right) E_m E_n \\
item_6 = \sum_{i,j,k,l,m,n} \left(e_{ij}e_{ji}\right)\left(e_{kl}e_{lm}e_{mk} + e_{km}e_{ml}e_{lk}\right) E_n + \\
\qquad \sum_{i,j,k,l,m,n} \left(e_{ij}e_{jk}e_{kl}e_{lm}e_{mi} + e_{im}e_{ml}e_{lk}e_{kj}e_{ji}\right) E_n \\
item_7 = \sum_{i,j,k,l,m,n} \left(e_{ij}e_{ji}\right)\left(e_{kl}e_{lm}e_{mn}e_{nk} + e_{kn}e_{nm}e_{ml}e_{lk}\right) + \\
\qquad \sum_{i,j,k,l,m,n} \left(e_{ij}e_{jk}e_{ki}\right)\left(e_{lm}e_{mn}e_{nl}\right) + \\
\qquad \sum_{i,j,k,l,m,n} \left(e_{ij}e_{ji}\right)\left(e_{kl}e_{lk}\right)\left(e_{mn}e_{nm}\right) + \\
\qquad \sum_{i,j,k,l,m,n} \left(e_{ij}e_{jk}e_{kl}e_{lm}e_{mn}e_{ni} + e_{in}e_{nm}e_{ml}e_{lk}e_{kj}e_{ji}\right)
\end{cases} . \tag{6}$$

The detailed explanation regarding the main characteristics of these seven terms can be found below:

$item_1 = \prod_{i=1}^{6} E_i$ This item denotes the interactions of six main enabler categories, namely, from E_1 to E_6

$item_2 = \prod_{i=1}^{6} e_{ii}$ According to the original constraint setting (i.e., no self-loop found in the enablers digraph), the resultant of this item is null

$item_3 = \sum_{i,j,k,l,m,n} \left(e_{ij}e_{ji}\right) E_k E_l E_m E_n$ Each component involved in this item stands for dual-element interdependency loop (denoted by $e_{ij}e_{ji}$) and the measure of the

	remaining four (in the context of FMS case) unconnected elements
$item_4 = \sum_{i,j,k,l,m,n} \big(e_{ij}e_{jk}e_{ki}$ $+e_{ik}e_{kj}e_{ji}\big)E_lE_mE_n$	In this term, a set of three-element interdependency loops (indicated by $e_{ij}e_{jk}e_{ki}$ and $e_{ik}e_{kj}e_{ji}$) and the measure of the remaining three (still in the context of FMS case) are represented by the components of the fourth item
$item_5$	This item is composed of two sub-items, i.e., $\sum_{i,j,k,l,m,n} \big(e_{ij}e_{ji}\big)\big(e_{kl}e_{lk}\big) \quad E_mE_n$ and $\sum_{i,j,k,l,m,n} \big(e_{ij}e_{jk}e_{kl}e_{li} + e_{il}e_{lk}e_{kj}e_{ji}\big)E_mE_n$. Two dual-element (represented by $e_{ij}e_{ji}$ and $e_{kl}e_{lk}$) interdependency loops and two FMS enabler categories (denoted by E_mE_n) are involved in the first sub-item; while the second sub-item consists of two four-element interdependency loops (i.e., $e_{ij}e_{jk}e_{kl}e_{li}$ and $e_{il}e_{lk}e_{kj}e_{ji}$) and two FMS enablers (i.e., E_mE_n)
$item_6$	There are also two sub-items contained in this item. The first sub-item (i.e., $\sum_{i,j,k,l,m,n} \big(e_{ij}e_{ji}\big)\big(e_{kl}e_{lm}e_{mk} + e_{km}e_{ml}e_{lk}\big)E_n$) is a product of three terms, namely, one dual-element interdependency loop (i.e., $e_{ij}e_{ji}$), two three-element interdependency loop (i.e., $e_{kl}e_{lm}e_{mk}$ and $e_{km}e_{ml}e_{lk}$), and one FMS enabler category (i.e., E_n). Following the alike fashion found in $item_5$, two terms (i.e., two five-element interdependency loops and one FMS enabler category) are multiplied to form $\sum_{i,j,k,l,m,n} \big(e_{ij}e_{jk}e_{kl}e_{lm}e_{mi} + e_{im}e_{ml}e_{lk}e_{kj}e_{ji}\big)E_n$
$item_7$	Four sub-items are organized to configure the seventh item. A product of one dual-element interdependency loop (i.e., $e_{ij}e_{ji}$) and two four-element interdependency loop (i.e., $e_{kl}e_{lm}e_{mn}e_{nk}$ and $e_{kn}e_{nm}e_{ml}e_{lk}$) can be found in the first sub-item, i.e., $\sum_{i,j,k,l,m,n} \big(e_{ij}e_{ji}\big) \big(e_{kl}e_{lm}e_{mn}e_{nk} + e_{kn}e_{nm}e_{ml}e_{lk}\big)$. Then two three-element interdependency loops are multiplied to form the second sub-item, i.e., $\sum_{i,j,k,l,m,n} \big(e_{ij}e_{jk}e_{ki}\big)\big(e_{lm}e_{mn}e_{nl}\big)$. Next, a multiplication of three two-element interdependency

loops is performed to obtain the third sub-item, i.e., $\sum_{i,j,k,l,m,n} \left(e_{ij}e_{ji}\right)\left(e_{kl}e_{lk}\right)\left(e_{mn}e_{nm}\right)$. Finally, two six-element interdependency loops (i.e., $e_{ij}e_{jk}e_{kl}e_{lm}e_{mn}e_{ni}$ and $e_{in}e_{nm}e_{ml}e_{lk}e_{kj}e_{ji}$) are included in the fourth sub-item of the seventh item

Once all 7 types of items have been successfully defined, one can simply substitute the element values of the earlier built FMS enablers' matrix (i.e., $\mathbf{F}^*$ in Sect. 4.2.2) for the just established permanent function, i.e., $f^{\mathbf{F}^*}_{permanent}$. After performing some basic algebraic and arithmetic operations, the final form of permanent function will turn into Eq. (7) as shown below (adapted from [51]):

$$\begin{aligned} f^{F^*}_{permanent} &= E_1E_2E_3E_4E_5E_6 \\ &\quad + [(e_{13}e_{31})E_2E_4E_5E_6 + (e_{45}e_{54})E_1E_2E_3E_6 + (e_{16}e_{61})E_2E_3E_4E_5 \\ &\quad + (e_{34}e_{43})E_1E_2E_5E_6] + [(e_{13}e_{36}e_{61})E_2E_4E_5 + (e_{35}e_{54}e_{43})E_1E_2E_6 \\ &\quad + (e_{24}e_{43}e_{32})E_1E_5E_6 + (e_{36}e_{64}e_{43})E_1E_2E_5 + (e_{31}e_{14}e_{43})E_2E_5E_6] \\ &\quad + [(e_{13}e_{31})(e_{45}e_{54})E_2E_6 + (e_{45}e_{54})(e_{16}e_{61})E_2E_3 + (e_{25}e_{54}e_{43}e_{32})E_1E_6 \\ &\quad + (e_{36}e_{65}e_{54}e_{43})E_1E_2 + (e_{31}e_{15}e_{54}e_{43})E_2E_6] \\ &\quad + [(e_{31}e_{16}e_{65}e_{54}e_{43})E_2 + (e_{61}e_{16})(e_{24}e_{43}e_{32})E_5] + [(e_{61}e_{16})(e_{25}e_{54}e_{43}e_{32})] \end{aligned} \tag{7}$$

4.2.4 Creating FIT Value Scale

In general, FIT can be regarded as an indicator of the smoothness degree when an organization transforming itself from one state (normally in near out-of-date condition) to another state (preferably in an up-to-date situation). Mathematically, FIT can be described through Eq. (8) (adapted from [51]):

$$\text{FIT} = f^{\mathbf{F}^*}_{permanent} = \text{Permanent function of enablers' matrix} \tag{8}$$

By computing the value of FIT, the feasibility of initiating the targeted transformation for any given company can thus be quantified. Since no negative term is included in Eq. (6) (see Sect. 4.2.3), we can conclude that the larger input numerical values of E_i and e_{ij} will always lead to an overall higher output value of FIT. In order to calculate the expected FIT value, the acquisition of the values of E_i and e_{ij} via the following ways is a must.

E_i's value Each enabler category's value is decided through treating each E_i as a subsystem and applying the GTA method to it accordingly. E_i's value is normally determined through analyzing the available system data and relative personnel opinions in an case study organization. In the case of where obtaining a quantitative value is not practicable, a ranked value judgment based on a scale of 1–10 is employed as a trade-off. Nevertheless, no matter which case comes in, the finally value of E_i is largely influenced by the acquirability of each individual building block's (i.e., e_{ij}) value

e_{ij}'s value Under each enabler category, there are various influencing constitutes (i.e., e_{ij}). As suggested by the author of [51], a scale of 1–5 can be assigned to each e_{ij} individually for overcoming the directly immeasurable issue

To summarize, the FIT value can be gained as follows: First, identifying various influencing constitutes under each enabler category. Then, visualizing the number of constitutes and their corresponding correlation via a digraph. Once the digraph is drawn up, the next step is to interpret it with the help of matrix expression. Finally, after all preparations are done, the permanent function is obtainable and the FIT value can therefore be calculated.

4.2.5 Comparison

By identifying suitable enablers, we can see that FIT is a very helpful measurement in assisting us with the feasibility assessment of different organizations in terms of transforming themselves to a new business operating scenario. As shown in Eq. (FIT $= f^{\mathbf{F}^*}_{permanent}$ = Permanent function of enablers' matrix), the value of permanent function (with respect to the enablers' matrix) determines the final FIT value. In other words, for any selected firms (say two for comparison purpose), their similarity degree is high (from the "suitability to transformation" viewpoint) if their digraphs are selfsame; similarly, their digraphs will be identical if the corresponding enablers' matrices are alike. Therefore, in order to perform the applicable conversion feasibility analysis, the establishment of the corresponding permanent function is a key. Built on these deductions, the identification set for an firm can be express via Eq. (9) [51]:

$$/Z_1/Z_2/Z_3/Z_4/Z_{51} + Z_{52}/Z_{61} + Z_{62}/ \tag{9}$$

As we have learned that each obtained permanent function is composed of a set of items (seven of them for FMS case), and each individual item also consists of a couple of sub-items, the identification set shown in Eq. (9) for FMS case is understandable in which the value of the ith item is denoted by Z_i, and Z_{ij} stands for the value of the jth sub-item embraced in the ith item. By substituting the values of E_i and e_{ij} for the corresponding item and sub-item found in the permanent function, the values of Z_i and Z_{ij} can be acquired. In the case of sub-item does not exist, then let Z_{ij} equal to Z_i, i.e., $Z_{ij} = Z_i$.

In practice, for any two organizations, it is fair to say that the main enabler categories and the various involved empowering elements under each category for transforming into the desired target is rare same. Accordingly, the comparison of two companies can be conducted by assessing the relevant similarity or dissimilarity coefficient. Such coefficient is normally based on the numerical values of the items/sub-items found in an permanent function. The range of similarity/dissimilarity coefficient falls within [0, 1], i.e., the value of similarity coefficient factor is set as 1 while 0 is assigned to dissimilarity coefficient factor if two firms share a high degree of similarity (or low degree of dissimilarity) during the course of the targeted transformation; in a similar vein, the similarity and dissimilarity coefficient factor is set as 0 and 1, respectively, in the case of two firms show a high degree of dissimilarity (or low degree of similarity) within the planned conversion phase. As such, the dissimilarity coefficient for any two comparable firms was proposed in the form of Eq. (10) [51]:

$$\begin{aligned} Coefficient_{dissimilarity} &= \left(\frac{1}{U}\right)\sum_{i,j}\lambda_{ij} \\ U = \text{maximum of } &\left[\sum_{i,j}\left|Z_{ij}\right| \text{ and } \sum_{i,j}\left|Z'_{ij}\right|\right]. \end{aligned} \tag{10}$$

where Z_{ij} and Z'_{ij} represents the values of the terms involved in the chosen two comparable organizations, and the value of λ_{ij} is defined as $\lambda_{ij} = \left|Z_{ij} - Z'_{ij}\right|$. As shown in Eq. (10), only the absolute difference between the values of terms is considered. Meanwhile, the similarity coefficient can also be computed via Eq. (11) [51]:

$$Coefficient_{similarity} = 1 - Coefficient_{dissimilarity} \tag{11}$$

By introducing Eqs. (10) and (11), the main advantage of feasibility assessment can thus be quantified. Through the comparison of the relevant values (e.g., Z_{ij}, Z'_{ij}, $Coefficient_{dissimilarity}$, and $Coefficient_{similarity}$), the strengths and weaknesses of a particular participated organized can thus be identified and the possibility of corresponding improvement can also be evaluated.

4.2.6 Summary

Through Sects. 4.2.1 to 4.2.5, a GTA based approach for evaluating an organization's "fitness" degree towards a desired transformation are particularized. To summarize, the major stages embraced in this methodology are outlined as follows:

- Pinpoint the suitable enabler categories.
- Visualize the enabler categories' digraph.

- Allocate the necessary empowering elements under each category.
- Build up an empowering elements' digraph with respect to each enabler category.
- Establish the variable permanent matrix.
- Compute the permanent function at each sub-system level.
- Develop the scenario dependent matrix for enabler categories' digraph.
- Figure the permanent function of the previous stage obtained matrix.
- Sort the different companies in ascending/descending order. The company with the highest FIT value enjoys the best opportunity of fulfilling the examined transformation.
- Acquire the identification set for each considered organization via Eq. (9).
- Compare the relevant similarity/dissimilarity coefficient between two organizations based on Eqs. (10) and (11).
- Diarize the obtained results for future or further analysis.

5 Experimental Study Stage-1: Classifying Enabler Category and Identifying Empowering Element

Once the theoretical foundation of the present study has been finely established, an instant work would be verifying whether the proposed FIT measurement can be successfully applied to our focal question. In order to achieve this goal, a set of enabler categories and their involving empowering elements are first identified via a thorough literature review and an intensive communications with our industrial partner. Following a similar manner discovered in [51], a further grouping operation is performed on the classified enabler categories and the identified empowering elements (see Table 1) so that their permanent function value is more computable.

5.1 Behavioral Enabler Category (E_1)

This enabler class relates to the top management who is responsible for the implementation of maintainomics in an organization. In general term, this is often referred to as the strategic planning process which embraces the establishment of a company's main goals and objectives, and the allocation of required resources to achieve them [111]. In practice, top management often takes charge of such process, although resources, products, consumers, and competitors are among various factors that are inspected during the process of strategic planning.

According to [112], there is a growing imperative for companies to be able to mine and process big data in order to improve competitiveness. As a result, these enablers are the first step to identify and remove adoption barriers. For example,

Table 1 Power plant maintainomics enabler categories and the involving empowering elements

Set no.	Enabler category	Empowering element	References
Set 1	Behavioral	1. Top management commitment	[2, 52, 53]
		2. Clear vision	[54, 55]
		3. Comprehension on communication revolution	[56]
		4. Stay competitive	[52]
		5. Team spirit and motivation	[57]
		6. Attainability of trained employee	[58–60]
Set 2	Non-behavioral	1. Availability of data collection, storage, and analytic choices	[9, 61–64]
		2. Availability of good e-maintenance architecture and platform	[9, 14, 37, 39, 65–68]
		3. Availability of usable asset self-identification	[3, 13, 60]
		4. Availability of good suppliers	[2]
Set 3	Financial	1. Funding direct from companies	[59, 69]
		2. Funding from private sectors	[11, 59, 69]
		3. Funding from government	[55, 60, 69, 70]
		4. Funding from international cooperation	[11, 69, 71]
		5. Financial incentives	[13]
Set 4	Methodological	1. Effective data mining technologies	[72–74]
		2. Online conditional monitoring	[75]
		3. Effective use of data collection standards	[76],[55]
		4. Unmanned inspection	[77–81]

(continued)

Table 1 (continued)

Set no.	Enabler category	Empowering element	References
Set 5	Operational	1. Power plant maintenance scheduling (PPMS) optimize techniques	[82–85]
		2. Autonomous and/or remote machine condition inspection	[86–90]
		3. Advanced machine learning and computational intelligence techniques for processing big data	[91, 92]
		4. Open-source technologies (e.g., Hadoop) and parallel programming model (e.g., MapReduce)	[93–98]
		5. Advanced sensor technology, e.g., RFID	[3, 99–102]
		6. Proper decision making regarding the machine condition	[103–107]
Set 6	Human and cultural	1. Data-centric enterprise organizational culture	[31, 59, 70]
		2. People and skills challenges	[59, 60]
		3. Capability of making better decisions	[108]
		4. Maintenance culture	[109]
		5. Culture clash between traditional business analysts, statisticians, data-application developers, and others	[110]

McKinsey Institute pointed out in a report [53] that if top managers can rethink the role of information in business and invest in better systems (or personnel) to dissect and interpret big data, they ought to gain customer insights, effect more accurate budgeting and better performance management. In a similar vein, [52] suggested that success in the data-driven economy (e.g., e-maintenance) need to access and analyze endless insights on their business. Trouble is, there is huge gaps exist between what organizations want to have and what they are able to do. As advocated in [55] that one of important things is to make sure you have clear vision on the definition being used by your organization, because big data is not just a single set of data, it is able to connect different sets of data together and therefore to create even more sets of information. In addition, several authors (e.g., [58, 59]) concluded that to educate themselves (e.g., plant executives, maintenance managers, and work planners) on how to manipulate and analyze data can help the organizations to make effective decisions. Also, to change attitudes toward the practicality of working with large data stores [57] and to comprehend communication revolution [56] (e.g., a surge in machine-to-machine communications, increased interaction via mobile devices, and massive using of tracking systems) are important, because most companies' internal IT functions aren't up to the job.

5.2 Non-behavioral Enabler Category (E_2)

This enabler group refers to those theoretical foundations that are necessary for supporting e-maintenance deployment and data analysis. Main characteristic of maintainomics is that maintenance information and access to related services can become ubiquitous and transparently available across the maintenance operations chain. For this purpose, researchers increasingly deal with a variety of research fields ranging from operation & maintenance engineering, to software engineering, information and communication systems, and business management [2]. For example, several comprehensive architectural frameworks for e-maintenance have been proposed, such as [14, 37, 65]. In addition, from a technological point of view, maintainomics is made-up from one or several networks with servers (e.g., Intra-Net), wireless technologies (e.g., ZigBee), databases (e.g., open systems architecture for condition based maintenance (OSA-CBM)), semantic data modelling (e.g., MapReduce programming model), smart sensors [e.g., micro electro mechanical systems (MEMS)], personnel mobility supporting (e.g., tablets), and many more. Also, within maintainomics, asset self-identification [e.g., radio frequency identification (RFID)] is a key factor [60]. Furthermore, suppliers are the key non-behavioral enabler as well, because they can develop, evolve, and maintain products and services [2].

5.3 Financial Enabler Category (E_3)

This enabler set deals with the economic aspects of maintainomics. Collecting maintenance data, analyzing, using and integrating them within different organizations, requires financial supporting. Indeed, the era of big data reshapes the innovation processes from companies to private sectors, government and international cooperation. For example, the PROTEUS project is a collaborative initiative for implementation of web-based e-maintenance centers in Europe [71]. The Dynamite project which is founded by EU as well focused on a set of methodologies and tools to support the e-maintenance processes, such as smart tags, common database schemas, and financial cost-efficiency assessment [60]. In 2012, the White house launched a $200 million initiative and pointed out that the big data research can help government eases their tasks somehow and thus reducing the problems faced [70]. In order to view the different information types on the same computer terminal, the machinery information management open system alliance (MIMOSA) worked closely with the international standards organization (ISO) for machine condition assessment [60]. Also, several private sectors (e.g., healthcare, manufacturing, and consumer products) worked towards big data research [59, 69]. On the other hand, data itself offers opportunities for the companies. According to [13], an increasing number of companies recognize that e-maintenance can be seen as a business opportunity rather than a cost center.

5.4 Methodological Enabler Category (E_4)

This enabler classification concerns about the physical requirements for the efficient collection of maintenance related data, and the use of various approaches for the efficient data analysis. Different kinds of smart data acquisition devices are commercial available in the market. By investing heavily, most of them are purchasable. Nevertheless, to reach the full potential of maintainomics, a careful selection of suitable equipments and the proper use of data analyzing techniques are two things that we need to think twice before taking action. Looking at computational intelligence (CI), a powerful tool for data mining, as a branch of artificial intelligence, the development of CI enjoys a tremendous prosperity during the past two decades [92]. By mimicking some nature sourced principles, CI algorithms can offer us the capability of digging out the hidden patterns and/or correlations from a biggish data set. The potential of big data cannot be interpreted in a meaningful way without the help of novel computational data mining algorithms.

5.5 Operational Enabler Category (E_5)

This enabler category is related to the question of how to deal with the challenges of very large datasets in order to improve asset reliability and reduce maintenance outages? Several authors (e.g., [9, 13, 60, 113]) pointed out that colleting, storing, and retrieving all maintenance data (e.g., vibration analysis, non-destructive testing results, and digital images) is critical to resolving an unplanned equipment failure. As a result, more companies are rethinking their planning, operations and IT functions to improve their maintenance processes through large data sets. This requires more efficient scheduling optimization techniques, autonomous and/or remote online monitoring systems (e.g., [87]), new open-source technologies (e.g., non-relational databases, distributed processing framework, and parallel programming model), mature wireless sensor tools (e.g., RFID), and proper decision making regarding the machine condition to work together. For example, different optimization methods (e.g., ant colony optimization [82], artificial bee colony [114], dynamic programming, [115] genetic algorithm [116], simulated annealing [117], and hybrid approaches [118]) have been proposed in the literature to solve maintenance scheduling problem of power systems. Ref. [102] pointed out that smart wireless technology can help operators to identify failed steam traps and leaks as early as possible. In addition, thanks to the new open-source technologies (such as Hadoop), the operators' ability to process large datasets has been significantly improved.

5.6 Human and Cultural Enabler Category (E_6)

The last category focuses on the "soft" factors that affect the adoption of maintainomics. Nowadays, scholars convinced that companies who gain deeper insights into the data can gain superior value in a competitive marketplace. In addition, they agreed that the technology issues are not the biggest barrier [59]. Instead, the majority initiatives have failed to deliver expected results due to those initiatives have ignored the human and cultural side of organizations. For example, recent economist intelligence unit report pointed out that people and skills challenges, process and organizational structure considerations, and cultural changes are the key factors associated with successfully implementing big data initiatives [59]. In a similar vein, report [108] emphasized that democratization of data-driven decision-making, making cross-functional teams big data strategy architects, and leveraging key data and departments are largely aspirational. Another sub-enabler affecting the rapidly of adoption of maintainomics in organization is maintenance culture, with the idea that creation of new maintenance practices. For example, ref. [109] reviewed the maintenance culture in Nigeria and highlighted that poor maintenance culture (e.g., unplanned maintenance services, poor user habits, and lack of awareness) affects the productivity of Nigeria power generation stations.

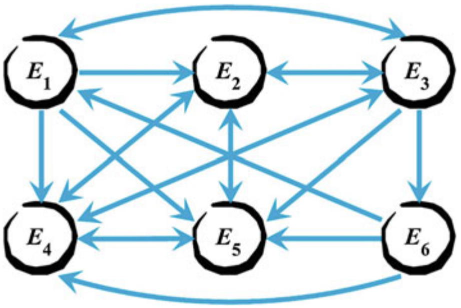


Fig. 3 Maintainomics scenario digraph representation

Finally, the different culture between developers, traditional business analysts, and systems administrators is also a crucial factor that affect the adoption of maintainomics.

5.7 Maintainomics Scenario Digraph Representation

According to the six main enabler categories and their associated interdependency, a maintainomics digraph can be drawn as depicted in Fig. 3:

Based on the derived maintainomics enabler categories' diagraph, the system level maintainomics matrix [see Sect. 4.2.2 for more details about its original form, i.e., Eq. (4)] can be re-written as Eq. (12):

$$\begin{gathered}
\mathbf{F} = \begin{array}{cc}
\begin{array}{c}\text{Enabler}\\ \text{Categories}\end{array} & \begin{array}{cccccc} E_1 & E_2 & E_3 & \cdots & \cdots & E_P \end{array} \\
\begin{array}{c} E_1 \\ E_2 \\ E_3 \\ \vdots \\ \vdots \\ E_P \end{array} & \begin{pmatrix} E_1 & e_{12} & e_{13} & \cdots & \cdots & e_{1P} \\ e_{21} & E_2 & e_{23} & \cdots & \cdots & e_{2P} \\ e_{31} & e_{32} & E_3 & \cdots & \cdots & e_{3P} \\ \vdots & \vdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ e_{P1} & e_{P2} & e_{P3} & \cdots & \cdots & E_P \end{pmatrix}
\end{array} \\
\Downarrow \\
\mathbf{F}^* = \begin{array}{cc}
\begin{array}{c}\text{Enabler}\\ \text{Categories}\end{array} & \begin{array}{cccccc} E_1 & E_2 & E_3 & E_4 & E_5 & E_6 \end{array} \\
\begin{array}{c} E_1 \\ E_2 \\ E_3 \\ E_4 \\ E_5 \\ E_6 \end{array} & \begin{pmatrix} E_1 & e_{12} & e_{13} & e_{14} & e_{15} & 0 \\ 0 & E_2 & e_{23} & e_{24} & e_{25} & 0 \\ e_{31} & e_{32} & E_3 & e_{34} & e_{35} & e_{36} \\ 0 & e_{42} & e_{43} & E_4 & e_{45} & 0 \\ 0 & e_{52} & 0 & e_{54} & E_5 & 0 \\ e_{61} & 0 & 0 & e_{64} & e_{65} & E_6 \end{pmatrix}
\end{array}
\end{gathered} . \tag{12}$$

6 Experimental Study Stage-2: FIT Value Calculation for Quantification Purpose

Obtaining a holistic digraph serves as a starting point for calculating the required FIT value that matches our maintainomics scenario. Apart from this, the following tasks also need to be performed.

Based on the identified empowering element under each enabler category, a set of digraphs are developed (as illustrated in Fig. 4a–f). Unlike the global digraph delineated in Fig. 3, the nodes in this local digraph stand for the empowering elements, while the edges are used to indicate their mutual relationship.

As shown in Fig. 4, according to the introduced measure scale, i.e., 1–10 for inheritance and 1–5 for interdependence, each enabler category's matrix and the corresponding permanent function value can also be obtained. A detailed breakdown of such computing process is demonstrated through $\mathbf{F}_1^*$ and $f_{permanent}^{\mathbf{F}_1^*}$.

Accordingly, the system level maintainomics matrix can be further re-written like Eq. (13) [see Sect. 5.7 for its intermediate form, i.e., Eq. (12)].

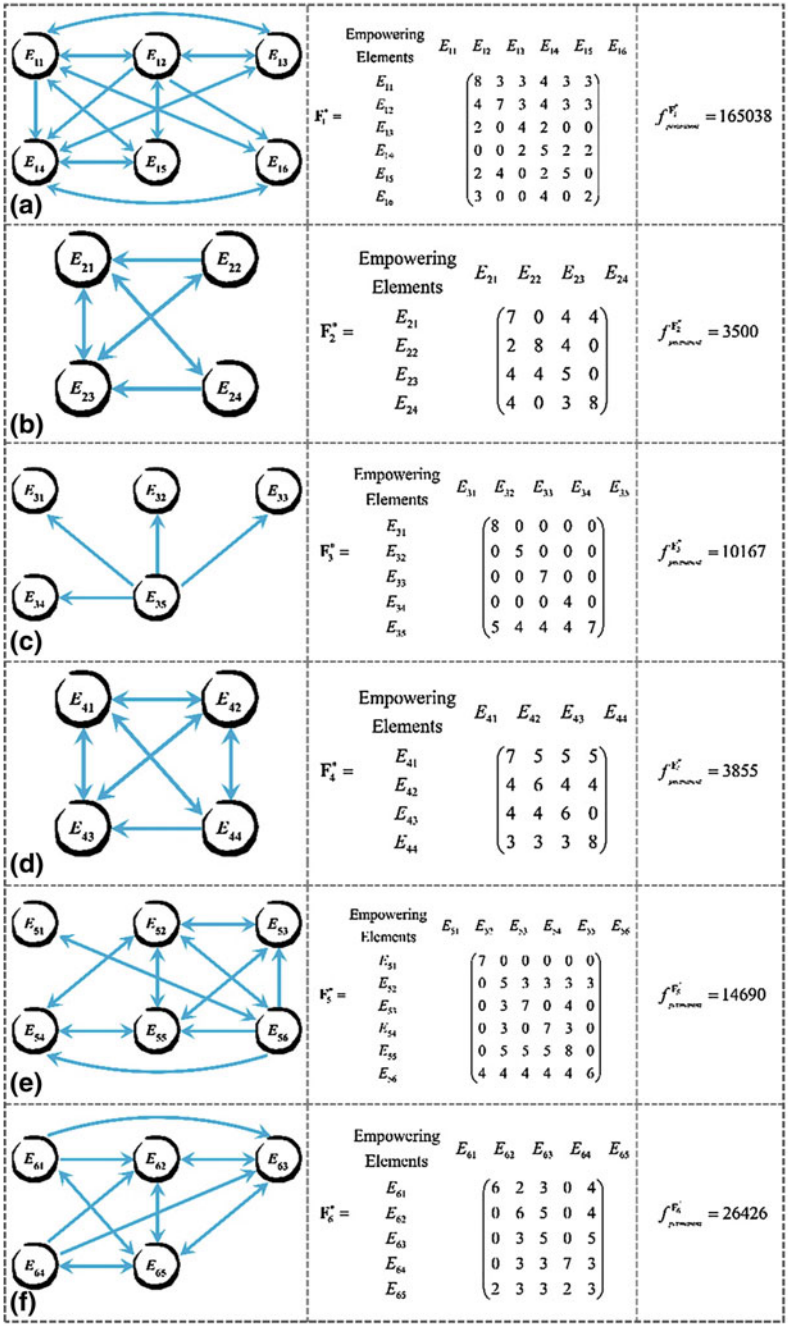
$$\mathbf{F}^* = \begin{array}{c|cccccc} \begin{array}{c}\text{Enabler}\\ \text{Categories}\end{array} & E_1 & E_2 & E_3 & E_4 & E_5 & E_6 \\ \hline E_1 & E_1 & e_{12} & e_{13} & e_{14} & e_{15} & 0 \\ E_2 & 0 & E_2 & e_{23} & e_{24} & e_{25} & 0 \\ E_3 & e_{31} & e_{32} & E_3 & e_{34} & e_{35} & e_{36} \\ E_4 & 0 & e_{42} & e_{43} & E_4 & e_{45} & 0 \\ E_5 & 0 & e_{52} & 0 & e_{54} & E_5 & 0 \\ E_6 & e_{61} & 0 & 0 & e_{64} & e_{65} & E_6 \end{array}$$

$$\Downarrow$$

$$\mathbf{F}^* = \begin{array}{c|cccccc} \begin{array}{c}\text{Enabler}\\ \text{Categories}\end{array} & E_1 & E_2 & E_3 & E_4 & E_5 & E_6 \\ \hline E_1 & 165038 & 2 & 2 & 4 & 3 & 0 \\ E_2 & 0 & 3500 & 4 & 4 & 3 & 0 \\ E_3 & 1 & 2 & 10167 & 4 & 3 & 3 \\ E_4 & 0 & 4 & 4 & 3855 & 4 & 0 \\ E_5 & 0 & 3 & 0 & 3 & 14690 & 0 \\ E_6 & 3 & 0 & 0 & 2 & 3 & 26426 \end{array} . \tag{13}$$

where the values of diagonal entries are replaced by just acquired sub-systems' permanent function values, namely, $E_1 = f_{permanent}^{\mathbf{F}_1^*} = 165038$, $E_2 = f_{permanent}^{\mathbf{F}_2^*} = 3500$, $E_3 = f_{permanent}^{\mathbf{F}_3^*} = 10167$, $E_4 = f_{permanent}^{\mathbf{F}_4^*} = 3855$, $E_5 = f_{permanent}^{\mathbf{F}_6^*} = 14690$, and $E_6 = f_{permanent}^{\mathbf{F}_6^*} = 26426$, while the corresponding scale value (1–5) are used to substitute the values of e_{ij}.

At this stage, the system level permanent function value, i.e., $f_{permanent}^{\mathbf{F}^*}$, is calculable which is found to be equal to 9.32018×10^{25}. Accordingly, the proposed



$$\mathbf{F}_1^* = \begin{array}{cc} \text{Empowering Elements} & \begin{array}{cccccc} E_{11} & E_{12} & E_{13} & E_{14} & E_{15} & E_{16} \end{array} \\ \begin{array}{c} E_{11} \\ E_{12} \\ E_{13} \\ E_{14} \\ E_{15} \\ E_{16} \end{array} & \begin{pmatrix} 8 & 3 & 3 & 4 & 3 & 3 \\ 4 & 7 & 3 & 4 & 3 & 3 \\ 2 & 0 & 4 & 2 & 0 & 0 \\ 0 & 0 & 2 & 5 & 2 & 2 \\ 2 & 4 & 0 & 2 & 5 & 0 \\ 3 & 0 & 0 & 4 & 0 & 2 \end{pmatrix} \end{array}$$

$$f_{permanent}^{\mathbf{F}_1^*} = 165038$$

$$\mathbf{F}_2^* = \begin{array}{cc} \text{Empowering Elements} & \begin{array}{cccc} E_{21} & E_{22} & E_{23} & E_{24} \end{array} \\ \begin{array}{c} E_{21} \\ E_{22} \\ E_{23} \\ E_{24} \end{array} & \begin{pmatrix} 7 & 0 & 4 & 4 \\ 2 & 8 & 4 & 0 \\ 4 & 4 & 5 & 0 \\ 4 & 0 & 3 & 8 \end{pmatrix} \end{array}$$

$$f_{permanent}^{\mathbf{F}_2^*} = 3500$$

$$\mathbf{F}_3^* = \begin{array}{cc} \text{Empowering Elements} & \begin{array}{ccccc} E_{31} & E_{32} & E_{33} & E_{34} & E_{35} \end{array} \\ \begin{array}{c} E_{31} \\ E_{32} \\ E_{33} \\ E_{34} \\ E_{35} \end{array} & \begin{pmatrix} 8 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 5 & 4 & 4 & 4 & 7 \end{pmatrix} \end{array}$$

$$f_{permanent}^{\mathbf{F}_3^*} = 10167$$

$$\mathbf{F}_4^* = \begin{array}{cc} \text{Empowering Elements} & \begin{array}{cccc} E_{41} & E_{42} & E_{43} & E_{44} \end{array} \\ \begin{array}{c} E_{41} \\ E_{42} \\ E_{43} \\ E_{44} \end{array} & \begin{pmatrix} 7 & 5 & 5 & 5 \\ 4 & 6 & 4 & 4 \\ 4 & 4 & 6 & 0 \\ 3 & 3 & 3 & 8 \end{pmatrix} \end{array}$$

$$f_{permanent}^{\mathbf{F}_4^*} = 3855$$

$$\mathbf{F}_5^* = \begin{array}{cc} \text{Empowering Elements} & \begin{array}{cccccc} E_{51} & E_{52} & E_{53} & E_{54} & E_{55} & E_{56} \end{array} \\ \begin{array}{c} E_{51} \\ E_{52} \\ E_{53} \\ E_{54} \\ E_{55} \\ E_{56} \end{array} & \begin{pmatrix} 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 3 & 3 & 3 & 3 \\ 0 & 3 & 7 & 0 & 4 & 0 \\ 0 & 3 & 0 & 7 & 3 & 0 \\ 0 & 5 & 5 & 5 & 8 & 0 \\ 4 & 4 & 4 & 4 & 4 & 6 \end{pmatrix} \end{array}$$

$$f_{permanent}^{\mathbf{F}_5^*} = 14690$$

$$\mathbf{F}_6^* = \begin{array}{cc} \text{Empowering Elements} & \begin{array}{ccccc} E_{61} & E_{62} & E_{63} & E_{64} & E_{65} \end{array} \\ \begin{array}{c} E_{61} \\ E_{62} \\ E_{63} \\ E_{64} \\ E_{65} \end{array} & \begin{pmatrix} 6 & 2 & 3 & 0 & 4 \\ 0 & 6 & 5 & 0 & 4 \\ 0 & 3 & 5 & 0 & 5 \\ 0 & 3 & 3 & 7 & 3 \\ 2 & 3 & 3 & 2 & 3 \end{pmatrix} \end{array}$$

$$f_{permanent}^{\mathbf{F}_6^*} = 26426$$

Fig. 4 Digraph representation for empowering elements

FIT measurement mathematically characterizes the feasibility of the chosen case in converting itself to the maintainomics scenario.

7 Experimental Study Stage-3: FIT Value Calculation for Comparison Purpose

In order to illustrate the usefulness of the proposed FIT measurement, a comparison study between two power plants is further introduced in this section in which the comparison approach is based on the methodology detailed in Sect. 4.2.5, while the selected power plant cases are briefed as follows:

- Power plant 1—fossil power plant: it is well known that fossil fuel-fired power plants are the main sources of power generating. Traditionally, power companies put “fail and fix” strategy into maintenance practices. However, as the pressure of rapid development around the globe, there is a need to continuously reduce unscheduled downtime and unexpected breakdowns.
- Power plant 2—wind power plant: several authors (e.g., [119–122]) convinced that wind power plants brings huge benefits for electricity generation due to they are clean, inexhaustible, and reduce the noise and visual disturbance to people. However, the wind industry is facing challenges with high installation and maintenance costs, and potentially longer time out of operation at failures [123, 124].

The required values for power plant 1 (PP_1) are already known through Sects. 5 and 6. Following the similar procedure, the values for power plant 2 (PP_2) are also obtainable. All values are listed below:

- PP_1: $E_1^{PP_1} = 165038$, $E_2^{PP_1} = 3500$, $E_3^{PP_1} = 10167$, $E_4^{PP_1} = 3855$, $E_5^{PP_1} = 14690$, $E_6^{PP_1} = 26426$, $e_{12}^{PP_1} = 2$, $e_{13}^{PP_1} = 2$, $e_{14}^{PP_1} = 4$, $e_{15}^{PP_1} = 3$, $e_{23}^{PP_1} = 4$, $e_{24}^{PP_1} = 4$, $e_{25}^{PP_1} = 3$, $e_{31}^{PP_1} = 1$, $e_{32}^{PP_1} = 2$, $e_{34}^{PP_1} = 4$, $e_{35}^{PP_1} = 3$, $e_{36}^{PP_1} = 3$, $e_{42}^{PP_1} = 4$, $e_{43}^{PP_1} = 4$, $e_{45}^{PP_1} = 4$, $e_{52}^{PP_1} = 3$, $e_{54}^{PP_1} = 3$, $e_{61}^{PP_1} = 3$, $e_{64}^{PP_1} = 2$, $e_{65}^{PP_1} = 3$,

$$\mathbf{F}^*_{PP_1} = \begin{array}{cc} \begin{array}{c}\text{Enabler}\\ \text{Categories}\end{array} & \begin{array}{cccccc} E_1 & E_2 & E_3 & E_4 & E_5 & E_6 \end{array} \\ \begin{array}{c} E_1 \\ E_2 \\ E_3 \\ E_4 \\ E_5 \\ E_6 \end{array} & \begin{pmatrix} 165038 & 2 & 2 & 4 & 3 & 0 \\ 0 & 3500 & 4 & 4 & 3 & 0 \\ 1 & 2 & 10167 & 4 & 3 & 3 \\ 0 & 4 & 4 & 3855 & 4 & 0 \\ 0 & 3 & 0 & 3 & 14690 & 0 \\ 3 & 0 & 0 & 2 & 3 & 26426 \end{pmatrix} \end{array},$$

  and $f_{permanent}^{\mathbf{F}^*_{PP_1}} = 9.32018 \times 10^{25}$.
- PP_2: $E_1^{PP_2} = 6863$, $E_2^{PP_2} = 1899$, $E_3^{PP_2} = 89689$, $E_4^{PP_2} = 5897$, $E_5^{PP_2} = 198996$, $E_6^{PP_2} = 9879$, $e_{12}^{PP_2} = 3$, $e_{13}^{PP_2} = 2$, $e_{14}^{PP_2} = 4$, $e_{15}^{PP_2} = 3$, $e_{16}^{PP_2} = 5$, $e_{24}^{PP_2} = 2$,

$$e_{25}^{PP_2} = 4,\quad e_{31}^{PP_2} = 2,\quad e_{32}^{PP_2} = 2,\quad e_{34}^{PP_2} = 4,\quad e_{35}^{PP_2} = 2,\quad e_{36}^{PP_2} = 2,\quad e_{43}^{PP_2} = 3,$$
$$e_{45}^{PP_2} = 4,\qquad e_{54}^{PP_2} = 5,\qquad e_{61}^{PP_2} = 4,\qquad e_{64}^{PP_2} = 4,\qquad e_{65}^{PP_2} = 3,$$

$$\mathbf{F}^*_{PP_2} = \begin{array}{c c} \begin{array}{c}\text{Enabler}\\ \text{Categories}\end{array} & \begin{array}{cccccc} E_1 & E_2 & E_3 & E_4 & E_5 & E_6 \end{array} \\ \begin{array}{c} E_1 \\ E_2 \\ E_3 \\ E_4 \\ E_5 \\ E_6 \end{array} & \begin{pmatrix} 6863 & 3 & 2 & 4 & 3 & 5 \\ 0 & 1899 & 0 & 2 & 4 & 0 \\ 1 & 2 & 89689 & 4 & 2 & 2 \\ 0 & 0 & 3 & 5897 & 4 & 0 \\ 0 & 0 & 0 & 5 & 198996 & 0 \\ 4 & 0 & 0 & 4 & 3 & 9879 \end{pmatrix} \end{array}, \text{ and}$$

$$f_{permanent}^{\mathbf{F}^*_{PP_2}} = 1.628 \times 10^{25}.$$

According to Eq. (10) (see Sect. 4.2.5), $U =$ maximum of $\left[\sum_{i,j} \left|Z_{ij}\right|\right.$ and $\sum_{i,j} \left|Z'_{ij}\right|] = f_{permanent}^{\mathbf{F}^*_{PP_1}} = 9.32018 \times 10^{25}$, and $\sum_{i,j} \lambda_{ij} = \sum_{i,j} \left|Z_{ij} - Z'_{ij}\right| = 7.69218 \times 10^{25}$, the dissimilarity coefficient $Coefficient_{dissimilarity}$ and the similarity coefficient $Coefficient_{similarity}$ equals to 0.78 and 0.22, respectively.

The results of this work indicate that the dissimilarity degree between PP_1 and PP_2 is high. In addition to this, although the overall permanent function value of PP_2 is lower than its competitor PP_1, PP_2 enjoys a high permanent function value for its methodological and operational enabler categories. On the contrary, PP_1 possesses a higher permanent function value for its behavioral enabler category which in turn implies the suitability for transition to maintainomics is largely influenced by the this enabler category and its associated empowering elements.

8 Future Work

In this chapter, a GTA-based FIT assessment methodology has been successfully applied to our focal problem, i.e., maintainomics transformation. Although the present work offers some novel contributions to both big data and maintenance literature, there is still room for further improvement. One immediately future research direction would be introducing fuzzy graph model. It is quite well acknowledged that graphs are simple models that are easy to use for representing interrelationships between objects in which objects are denoted by vertices, while the correlations are indicated by edges. Nevertheless, in spite of its convenience, when it comes to the situation of vagueness contained in the objects' description, or found in the relationships, or in both, traditional GTA often performs poorly [125]. The numerical values assigned to E_i and e_{ij}, respectively, in this study are crisp and deterministic in nature. With the ever increasing complexity of many modern systems (e.g., maintainomics case), the uncertainty gradually plays a pivotal role in any attempts of trying to optimize and maximize the overall system model's usefulness. Under such circumstances, the introduction and application of fuzzy relations to maintainomics FIT evaluation becomes necessary and important. Once the feasibility of the maintainomics is confirmed, another future research direction will be bringing more CI methods into e-maintenance operational level for dealing with big data analytics.

9 Conclusion

Thanks to the use of sensors in maintenance (e.g., online monitoring, smart meters, etc.), the rate of growth of generated sensory data far outstrips human capacity to consume it. As a result, enterprises increasingly meet the challenges imposed by how to convert myriads of available raw data to high-level actionable information. This book chapter studies such emerging phenomena and builds a set of enablers to address those issues. The end-product of this chapter is maintainomics, a new concept that facilitates the entry of e-maintenance to any enterprises by evaluating a permanent feasibility function obtained from an enablers' digraph. The experimental studies demonstrated the suitability of our proposed methodology.

References

1. Muller, A., Suhner, M.-C., Iung, B.: Maintenance alternative integration to prognosis process engineering. J. Qual. Maint. Eng. **13**, 198–211 (2007)
2. Kajko-Mattsson, M., Karim, R., Mirjamdotter, A.: Essential components of e-maintenance. Int. J. Perform. Eng. **7**, 555–571 (2011)
3. Haider, A., Koronios, A.: e-prognostics: a step towards e-maintenance of engineering assets. J. Theor. Appl. Electron. Commer. Res. **1**, 42–55 (2006)
4. Lee, J., Ni, J., Djurdjanovic, D., Qiu, H., Liao, H.: Intelligent prognostics tools and e-maintenance. Comput. Ind. **57**, 476–489 (2006)
5. Campos, J.: Development in the application of ICT in condition monitoring and maintenance. Comput. Ind. **60**, 1–20 (2009)
6. Jardine, A.K.S., Lin, D., Banjevic, D.: A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mech. Syst. Signal Process. **20**, 1483–1510 (2006)
7. Niu, G., Yang, B.-S., Pecht, M.: Development of an optimized condition-based maintenance system by data fusion and reliability-centered maintenance. Reliab. Eng. Syst. Saf. **95**, 786–796 (2010)
8. Iung, B.: From remote maintenance to MAS-based e-maintenance of an industrial process. J. Intell. Manuf. **14**, 59–82 (2003)
9. Levrat, E., Iung, B., Marquez, A.C.: e-maintenance: a review and conceptual framework. Prod. Plann. Control **19**, 408–429 (2008)
10. Muller, A., Marquez, A.C., Iung, B.: On the concept of e-maintenance: review and current research. Reliab. Eng. Syst. Saf. **93**, 1165–1187 (2008)
11. Karim, R., Candell, O., Söderholm, P.: e-maintenance and information logistics: aspects of content format. J. Qual. Maint. Eng. **15**, 308–324 (2009)

12. Karim, R., Söderholm, P., Candell, O.: Development of ICT-based maintenance support services. J. Qual. Maint. Eng. **15**, 127–150 (2009)
13. Jantunen, E., Emmanouilidis, C., Arnaiz, A., Gilabert, E.: e-maintenance: trends, challenges and opportunities for modern industry. In: Proceedings of the 18th IFAC World Congress, pp. 453-458. Milano, Italy, 28 Aug–02 Sept 2011
14. Han, T., Yang, B.-S.: Development of an e-maintenance system integrating advanced techniques. Comput. Ind. **57**, 569–580 (2006)
15. Choi, J.-B., Yeum, S.-W., Ko, H.-O., Kim, Y.-J., Kim, H.-K., Choi, Y.-H., et al.: Development of a web-based aging monitoring system for an integrity evaluation of the major components in a nuclear power plant. Int. J. Press. Vessels Pip. **87**, 33–40 (2010)
16. Vo, C.C., Chilamkurti, N., Loke, S.W., Torabi, T.: Radio-Mama_an RFID based business process framework for asset management. J. Netw. Comput. Appl. **34**, 990–997 (2011)
17. Miertschin, K.W., Forrest, B.D.: Analysis of Tobyhanna army depot's radio frequency identification (RFID) pilot program: RFID as an asset management tool. Master thesis, Naval Postgraduate School, Monterey, CA, USA, 2005
18. Mahakul, T.K., Baboo, S., Patnaik, S.: Implementation of enterprise asset management using IT tools: a case study of IB thermal power station. J. Inf. Technol. Manage. **XVI**, 39–67 (2005)
19. Briones, L., Bustamante, P., Serna, M.A.: Robicen: a wall-climbing pneumatic robot for inspection in nuclear power plants. Robot. Comput. Integr. Manuf. **11**, 287–292 (1994)
20. Balaguer, C., Gimenez, A., Jardon, A.: Climbing robots' mobility for inspection and maintenance of 3D complex environments. Auton. Robots **18**, 157–169 (2005)
21. Luk, B.L., Cooke, D.S., Galt, S., Collie, A.A., Chen, S.: Intelligent legged climbing service robot for remote maintenance applications in hazardous environments. Robot. Auton. Syst. **53**, 142–152 (2005)
22. Wang, W., Wang, K., Zhang, H.: Crawling gait realization of the mini-modular climbing caterpillar robot. Prog. Nat. Sci. **19**, 1821–1829 (2009)
23. Lim, J., Park, H., An, J., Hong, Y.-S., Kim, B., Yi, B.-J.: One pneumatic line based inchworm-like micro robot for half-inch pipe inspection. Mechatronics **18**, 315–322 (2008)
24. Neubauer, W.: A spider-like robot that climbs vertically in ducts or pipes. In: Presented at the Proceedings of IEEE/RSJ International Conference on Intelligent Robotics and Systems, pp. 1178-1185, 1994
25. Prasad, E.N., Kannan, M., Azarudeen, A., Karuppasamy, N.: Defect identification in pipe lines using pipe inspection robot. Int. J. Mech. Eng. Robot. Res. **1**, 20–31 (2012)
26. Hu, Z., Appleton, E.: Dynamic characteristics of a novel self-drive pipeline pig. IEEE Trans. Rob. **21**, 781–789 (2005)
27. Schmidt, D., Berns, K.: Climbing robots for maintenance and inspections of vertical structures—a survey of design aspects and technologies. Robot. Auton. Syst. **61**, 1288–1305 (2013)
28. Roslin, N.S., Anuar, A., Jalal, M.F.A., Sahari, K.S.M.: A review: hybrid locomotion of in-pipe inspection robot. Proc. Eng. **41**, 1456–1462 (2012)
29. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., et al.: Big Data: The Next Frontier for Innovation, Competition, and Productivity. McKinsey Global Institute, Washington (2011)
30. O'Driscoll, A., Daugelaite, J., Sleator, R.D.: 'Big data', Hadoop and cloud computing in genomics. J. Biomed. Inform. **46**, 774–781 (2013)
31. Chang, R.M., Kauffman, R.J., Kwon, Y.: Understanding the paradigm shift to computational social science in the presence of big data. Decis. Support Syst. **63**, 67–80 (2014)
32. Wu, X., Zhu, X., Wu, G.-Q., Ding, W.: Data mining with big data. IEEE Trans. Knowl. Data Eng. **26**, 97–107 (2014)
33. Dai, J., Huang, J., Huang, S., Liu, Y., Sun, Y.: The Hadoop stack: new paradigm for big data storage and processing. Int. Technol. J. **16**, 92–110 (2012)
34. Sagiroglu, S., Sinanc, D.: Big data: a review. In: Presented at the 2013 International Conference on Collaboration Technologies and Systems (CTS), pp. 42-47, 2013

35. Nieva, T., Wegmann, A.: A conceptual model for remote data acquisition systems. Comput. Ind. **47**, 215–237 (2002)
36. Kans, M., Ingwald, A.: Common database for cost-effective improvement of maintenance performance. Int. J. Prod. Econ. **113**, 734–747 (2008)
37. Pistofidis, P., Emmanouilidis, C., Koulamas, C., Karampatzakis, D., Papathanassiou, N.: A layered e-maintenance architecture power by smart wireless monitoring components. In: Proceedings of the 2012 International Conference on Industrial Technology (ICIT), pp. 1-6, 2012
38. Iung, B., Marquez, A.C.: Special issue on e-maintenance. Comput. Ind. **57**, 473–475 (2006)
39. Iung, B., Levrat, E., Marquez, A.C., Erbe, H.: Conceptual framework for e-maintenance: illustration by e-maintenance technologies and platforms. Ann. Rev. Control **33**, 220–229 (2009)
40. Yao, Y.: A spectrum decision support system for cognitive radio networks. Licentiate Thesis, School of Computing, Blekinge Institute of Technology, Karlskrona, Sweden, 2012
41. Marais, K.B., Saleh, J.H.: Beyond its cost, the value of maintenance: an analytical framework for capturing its net present value. Reliab. Eng. Syst. Saf. **94**, 644–657 (2009)
42. Kalafatas, G.: A graph theoretic modeling framework for generalized transportation systems with congestion phenomena. PhD thesis, Purdue University, West Lafayette, Indiana, 2010
43. Thakkar, J., Kanda, A., Deshmukh, S.G.: Evaluation of buyer-supplier relationships using an integrated mathematical approach of interpretive structural modeling (ISM) and graph theoretic matrix: the case study of Indian automotive SMEs. J. Manuf. Technol. Manage. **19**, 92–124 (2008)
44. Franceschet, M., Gubiani, D., Montanari, A., Piazza, C.: A graph-theoretic approach to map conceptual designs to XML schemas. ACM Trans Database Syst **38**, 6:1–6:44 (2013)
45. Sabharwal, S., Garg, S.: Determining cost effectiveness index of remanufacturing: a graph theoretic approach. Int. J. Prod. Econ. **144**, 521–532 (2013)
46. Hou, F., Shen, W.-M.: Graph-based optimal reconfiguration planning for self-reconfigurable robots. Robot. Auton. Syst. **62**, 1047–1059 (2014)
47. Pishvaee, M.S., Rabbani, M.: A graph theoretic-based heuristic algorithm for responsive supply chain network design with direct and indirect shipment. Adv. Eng. Softw. **42**, 57–63 (2011)
48. Even, S., Even, G.: Graph Algorithms, 2nd edn. Cambridge University Press, New York (2012). ISBN 978-0-521-51718-8
49. Mesbahi, M., Egerstedt, M.: Graph Theoretic Methods in Multiagent Networks. Princeton University Press, Princeton (2010). ISBN 978-0-691-14061-2
50. Kreyszig, E., Kreyszig, H., Norminton, E.J.: Advanced Engineering Mathematics, 10th edn. Wiley, Hoboken (2011). ISBN 978-0-470-45836-5
51. Raj, T., Shankar, R., Suhaib, M.: GTA-based framework for evaluating the feasibility of transition to FMS. J. Manuf. Technol. Manage. **21**, 160–187 (2010)
52. Johnson, J.E.: Big data + big analytics = big opportunity. Financ. Executive **28**, 50–53 (2012)
53. Brynjolffson, E., Hammerbacher, J., Stevens, B.: Competing Through Data: Three Experts Offer Their Game Plans. McKinsey Global Institute, Washington (2011)
54. Buhl, H.U., Röglinger, M., Moser, F., Heidemann, J.: Big data: a fashionable topic with(out) sustainable relevance for research and practice? Bus. Inf. Syst. Eng. **2**, 65–69 (2013)
55. Jackson, R.A.: Big data big risk. Internal Auditor **70**, 34–38 (2013)
56. Big data needn't be a big headache: how to tackle mind-blowing amounts of information. Strateg. Dir. **28**, 22–24 (2012)
57. Alstyne, M.V., Brynjolfsson, E., Madnick, S.: Why not one big database: principles for data ownership. Decis. Support Syst. **15**, 267–284 (1995)
58. Chen, H., Chiang, R.H.L., Storey, V.C.: Business intelligence and analytics: from big data to big impact. MIS Q. **36**, 1165–1188 (2012)

59. Watson, J.: Big Data and Consumer Products Companies: People, Processes and Culture Barriers. The Economist Intelligence Unit Limited, London, New York, Hong Kong, Geneva (2013)
60. Emmanouilidis, C., Jantunen, E., Gilabert, E., Arnaiz, A., Starr, A.: e-maintenance update: the road to success for modern industry. In: Proceedings of the 24th International Congress on Condition Monitoring and Diagnostics Engineering Management, pp. 1–10. Stavanger, Norway, ISBN 0-9541307-2-3, 2011
61. Computer_Weekly. (2013) Big data storage choices. Computer Weekly 1–3
62. Huang, Y.-S., Duy, D., Fang, C.-C.: Efficient maintenance of basic statistical functions in data warehouses. Decis. Support Syst. **57**, 94–104 (2014)
63. Xu, X.-B., Yang, Z.-Q., Xiu, J.-P., Liu, C.: A big data acquisition engine based on rule engine. J. China Univ. Posts Telecommun. **20**, 45–49 (2013)
64. Urbani, J., Kotoulas, S., Maassen, J., Harmelen, F.V., Bal, H.: WebPIE: a Web-scale parallel inference engine using MapReduce. Web Semant.: Sci. Serv. Agents World Wide Web **10**, 59–75 (2012)
65. Candell, O., Karim, R., Parida, A.: Development of information system for e-maintenance solutions within the aerospace industry. Int. J. Perform. Eng. **7**, 583–592 (2011)
66. Chebel-Morello, B., Medjaher, K., Arab, A.H., Bandou, F., Bouchaib, S., Zerhouni, N.: e-maintenance for photovoltaic power generation system. Energy Proc. **18**, 640–643 (2012)
67. Hung, M.-H., Chen, K.-Y., Ho, R.-W., Cheng, F.-T.: Development of an e-diagnostics/maintenance framework for semiconductor factories with security considerations. Adv. Eng. Inform. **17**, 165–178 (2003)
68. Yu, R., Iung, B., Panetto, H.: A multi-agents based e-maintenance system with case-based reasoning decision support. Eng. Appl. Artif. Intell. **16**, 321–333 (2003)
69. Katal, A., Wazid, M., Goudar, R.H.: Big data: issues, challenges, tools and good practices. In: Presented at the 2013 Sixth International Conference on Contemporary Computing (IC3), pp. 404–409, 2013
70. Gobble, M.M.: Big data: the next big thing in innovation. Res. Technol. Manage. **56**, 64–66 (2013)
71. Hausladen, I., Bechheim, C.: e-maintenance platform as a basis for business process integration. In: Proceedings of the 2nd IEEE International Conference on Industrial Informatics (INDIN), pp. 46–51, 2004
72. Crowe, J., Candlish, J.R.: Data analytics: the next big thing in information. Grey J. **9**, 157–159 (2013)
73. Lee, J., Lapira, E., Bagheri, B., Kao, H.-A.: Recent advances and trends in predictive manufacturing systems in big data environment. Manuf. Lett. **1**, 38–41 (2013)
74. Pedrycz, W.: Granular Computing: Analysis and Design of Intelligent Systems. CRC Press, Boca Raton (2013). ISBN 978-1439886816
75. Yam, R.C.M., Tse, P.W., Li, L., Tu, P.: Intelligent predictive decision support system for condition-based maintenance. Int. J. Adv. Manuf. Technol. **17**, 383–391 (2001)
76. Cumbley, R., Church, P.: Is “Big Data” creepy. Comput. Law Secur. Rep. **29**, 601–609 (2013)
77. Bloss, R.: By air, land and sea, the unmanned vehicles are coming. Ind. Robot **34**, 12–16 (2007)
78. Pagnano, A., Höpf, M., Teti, R.: A roadmap for automated power line inspection. Maintenance and repair. Proc. CIRP **12**, 234–239 (2013)
79. Jones, D.I., Earp, G.K.: Camera sightline pointing requirements for aerial inspection of overhead power lines. Electr. Power Syst. Res. **57**, 73–82 (2001)
80. Wan, S., Bian, X., Chen, L., Yu, D., Wang, L., Guan, Z.: Electrostatic discharge effect on safe distance determination for 500 kV ac power line’s helicopter inspection. J. Electrostat. **71**, 778–780 (2013)
81. Jones, D.I., Whitworth, C.C., Earp, G.K., Duller, A.W.G.: A laboratory test-bed for an automated power line inspection system. Control Eng. Pract. **13**, 835–851 (2005)

82. Foong, W.K.: Ant colony optimization for power plant maintenance scheduling. Unpublished doctoral thesis, School of Civil and Environmental Engineering, University of Adelaide, 2007
83. Foong, W.K., Maier, H.R., Simpson, A.R.: Power plant maintenance scheduling using ant colony optimization. In: Chan, F.T.S., Tiwari, M.K. (eds.) Swarm Intelligence, Focus on Ant and Particle Swarm Optimization. Chapter 6, pp. 289–320. InTech, Vienna (2007). ISBN 978-3-902613-09-7
84. Foong, W.K., Simpson, A.R., Maier, H.R., Stolp, S.: Ant colony optimization for power plant maintenance scheduling optimization—a five-station hydropower system. Ann. Oper. Res. **159**, 433–450 (2008)
85. Mohanta, D.K., Sadhu, P.K., Chakrabarti, R.: Deterministic and stochastic approach for safety and reliability optimization of captive power plant maintenance scheduling using GA/SA-based hybrid techniques. Reliab. Eng. Syst. Saf. **92**, 187–199 (2007)
86. Netland, Y., Skavhaug, A.: Two pilot experiments on the feasibility of telerobotic inspection of offshore wind turbines. Presented at the 2nd Mediterranean Conference on Embedded Computing (MECD—2013, ECyPS'2013), Budva, Montenegro, pp. 1–4, 2013
87. Pan, M.-C., Li, P.-C., Cheng, Y.-R.: Remote online machine condition monitoring system. Measurement **41**, 912–921 (2008)
88. Das, J.D., S. Chowdhuri, J. Bera, and G. Sarkar, "Remote monitoring of different electrical parameters of multi-machine system using PC," *Measurement,* vol. 45, pp. 118-125, 2012
89. Mendoza-Jasso, J., Ornelas-Vargas, G., Castañeda-Miranda, R., Ventura-Ramos, E., Zepeda-Garrido, A., Herrera-Ruiz, G.: FPGA-based real-time remote monitoring system. Comput. Electron. Agric. **49**, 272–286 (2005)
90. Jonsson, K., Holmström, J., Levén, P.: Organizational dimensions of e-maintenance: a multi-contextual perspective. Int. J. Syst. Assur. Eng. Manage. **1**, 210–218 (2010)
91. Witten, I.H., Frank, E., Hall, M.A.: Data mining: practical machine learning tools and techniques, 3rd edn. Morgan Kaufmann, Elsevier Inc., Burlington (2011)
92. Xing, B., Gao, W.-J.: Innovative computational intelligence: a rough guide to 134 clever algorithms. Springer International Publishing Switzerland, Cham, Heidelberg, New York, Dordrecht, London, ISBN 978-3-319-03403-4 (2014)
93. Dai, J., Huang, J., Huang, S., Liu, Y., Sun, Y.: The Hadoop stack: new paradigm for big data storage and processing. Int. Technol. J. **16**, 92–110 (2012)
94. Patel, A.B., Birla, M., Nair, U.: Addressing big data problem using Hadoop and map reduce. In: Proceedings of 2012 Nirma University International Conference on Engineering (NUiCONE), pp. 1–5, 06–08 Dec 2012
95. Qiu, Z., Lin, Z.-W., Ma, Y.: Research of Hadoop-based data flow management system. J. China Univ. Posts Telecommun. **18**, 164–168 (2011)
96. Edwards, M., Rambani, A., Zhu, Y., Musavi, M.: Design of Hadoop-based framework for analytics of large synchrophasor datasets. Procedia Computer Science **12**, 254–258 (2012)
97. ElSheikh, G., ElNainay, M.Y., ElShehaby, S., Abougabal, M.S.: SODIM: service oriented data integration based on MapReduce. Alexandria Eng. J. **52**, 313–318 (2013)
98. Kolberg, W., Marcos, P.D.B., Anjos, J.C.S., Miyazaki, A.K.S., Geyer, C.R., Arantes, L.B.: MRSG—a MapReduce simulator over SimGrid. Parallel Comput. **39**, 233–244 (2013)
99. Ko, C.-H.: RFID-based building maintenance system. Autom. Constr. **18**, 275–284 (2009)
100. Osman, M.S., Ram, B., Stanfield, P., Samanlioglu, F., Davis, L., Bhadury, J.: Radio frequency identification system optimisation models for lifecycle of a durable product. Int. J. Prod. Res. **48**, 2699–2721 (2010)
101. Xing, B., Gao, W.-J., Marwala, T.: The applications of computational intelligence in radio frequency identification research. In: IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC), pp. 2067–2072. Seoul, Korea, 14–17 Oct 2012
102. MacDonald, I.: Smart wireless provides a sound basis for improved performance. Mod. Power Syst. 20–22 (2012)

103. Herzog, M.A.: Machine and component residual life estimation through the application of neural networks. Master thesis, Department of Mechanical and Aeronautical Engineering, Faculty of Engineering, University of Pretoria, Pretoria, South Africa, 2006
104. Herzog, M.A., Marwala, T., Heyns, P.S.: Machine and component residual life estimation through the application of neural networks. Reliab. Eng. Syst. Saf. **94**, 479–489 (2009)
105. Marwala, T.: Fault identification using neural networks and vibration data. Unpublished doctoral thesis, St. John's College, University of Cambridge, 2000
106. Xing, B., Gao, W.-J.: Computational Intelligence in Remanufacturing. IGI Global, Hershey (2014). ISBN 978-1-4666-4908-8
107. Gao, W.-J., Xing, B., Marwala, T.: Teaching—learning-based optimization approach for enhancing remanufacturability pre-evaluation system's reliability. In: IEEE Symposium Series on Computational Intelligence (IEEE SSCI), pp. 235-239. Singapore, 15–19 Apr 2013
108. Whelan, C.: Big data and the democratisation of decisions. The Economist Intelligence Unit Limited, London, New York, Hong Kong, Geneva (2012)
109. Onohaebi, O.S., Lawai, V.O.: Poor maintenance culture; the bane to electric power generation in Nigeria. J. Econ. Eng. 28–33 (2010)
110. Manoochehri, M.: Data just right: introduction to large-scale data and analytics. Pearson Education Inc, Upper Saddle River (2014). ISBN 978-0-321-89865-4
111. Pride, W.M., Hughes, R.J., Kapoor, J.R.: Business, 12th edn. South-Western, Cengage Learning, Mason (2014). 2014
112. Brown, B., Chui, M., Manyika, J.: Are you ready for the era of 'big data'?. McKinsey Global Institute, New York (2011)
113. Al-Qahtani, M.S., Aramco, S.: Information and communication technology infrastructure in e-maintenance. In: Proceedings of the Fourth International Conference on Information, Process, and Knowledge Management, ISBN 978-1-61208-181-6, pp. 7–11, 2012
114. Anandhakumar, R., Subramanian, S., Ganesan, S.: Modified ABC algorithm for generator maintenance scheduling. Int. J. Comput. Electr. Eng. **3**, 812–819 (2011)
115. Moghaddam, K.S., Usher, J.S.: Preventive maintenance and replacement scheduling for repairable and maintainable systems using dynamic programming. Comput. Ind. Eng. **60**, 654–665 (2011)
116. Wang, Y., Handschin, E.: A new genetic algorithm for preventive unit maintenance scheduling of power systems. Electr. Power Energy Syst. **22**, 343–348 (2000)
117. Saraiva, J.T., Pereira, M.L., Mendes, V.T., Sousa, J.C.: A simulated annealing based approach to solve the generator maintenance scheduling problem. Electr. Power Syst. Res. **81**, 1283–1291 (2011)
118. Yare, Y., Venayagamoorthy, G.K.: Optimal maintenance scheduling of generators using multiple swarms-MDPSO framework. Eng. Appl. Artif. Intell. **23**, 895–910 (2010)
119. Breton, S.P., Moe, G.: Status, plans and technologies for offshore wind turbines in Europe and North America. Renew. Energy **34**, 646–654 (2009)
120. Akdag, S.A., Dinler, A.: A new method to estimate weibull parameters for wind energy applications. Energy Convers. Manag. **50**, 1761–1766 (2009)
121. Irfan, U., Qamar-uz-Zaman, C., Andrew, J.C.: An evaluation of wind energy potential at Kati Bandar, Pakistan. Renew. Sustain. Energy Rev. **14**, 856–861 (2010)
122. Esteban, M.D., Diez, J.J., Lpez, J.S., Negro, V.: Why offshore wind energy? Renew. Energy **36**, 444–450 (2011)
123. Nguyen, T.H., Prinz, A., Friisø, T., Nossum, R., Tyapin, I.: A framework for data integration of offshore wind farms. Renew. Energy **60**, 150–161 (2013)
124. Hameed, Z., Vatn, J., Heggset, J.: Challenges in the reliability and maintainability data collection for offshore wind turbines. Renew. Energy **36**, 2154–2165 (2011)
125. Sunitha, M.S.: Studies on fuzzy graphs. Doctoral thesis, Department of Mathematics, Faculty of Science, Cochin University of Science and Technology, Cochin, 2001

Incrementally Mining Frequent Patterns from Large Database

Yue-Shi Lee and Show-Jane Yen

Abstract Mining *frequent patterns* is an important task in *data mining* area, which is to find the itemsets frequently purchased together from a transaction database. However, the transactions will grow rapidly, such that the size of the transaction database becomes bigger and bigger due to the addition of the new transactions. The users may eager for getting the latest frequent patterns from the large database as soon as possible in order to make the best decision. Therefore, it has become an important issue to propose an efficient method for finding the latest frequent patterns when the transactions keep being added into the database. Although tree-based approaches have been recently adopted in most of the studies in this field, they have to re-scan the original database and generate a large tree structure. In this paper, we propose two efficient algorithms which only keep frequent items in a condensed tree structure. When a set of new transactions is added into the database, our algorithms can efficiently update the tree structure without scanning the original database.

Keywords Data mining · Frequent pattern · Incremental mining · Tree structure · Transaction database

Y.-S. Lee · S.-J. Yen (✉)
Department of Computer Science and Information Engineering, Ming Chuan University,
Taoyuan, Taiwan
e-mail: sjyen@mail.mcu.edu.tw

Y.-S. Lee
e-mail: leeys@mail.mcu.edu.tw

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_6,

1 Introduction

With the improvement of hardware, computers now can store, calculate, analyze, and sort a great number of data to figure out useful ones for us. The term "*Data Mining*" , occurring to detect what the useful information for humans is, is used to discover the unknown but possibly useful information from a large dataset. *Mining frequent patterns* [1–6] is one of the important topics in data mining, which is used to analyze the consumers' behaviors to examine if they tend to buy a certain product and another certain product at the same time. From the viewpoint of marketing, companies can easily attract their customers with sales promotion or recommendations and therefore can compound profits.

The definitions about mining frequent patterns are described as follows [1]. A *transaction database* consists of a set of transactions, in which each transaction includes a transaction-id (TID) and the items purchased in this transaction. Table 1 is an example of a transaction database. Let $I = \{i_1, i_2, \ldots, i_n\}$. be the set of all items. An *itemset* is a subset of *I*, which is denoted as $\{a_1, a_2, \ldots, a_m\}$ ($\forall a_i \in I.$, $1 \le i \le m$, $1 \le m \le n$). A transaction in the database *supports* an itemset if the itemset is a subset of the items in the transaction. The *support for an itemset* is the fraction of total transactions which support the itemset. Given a user-specified *minimum support* threshold, the *frequent itemsets* are the itemsets whose supports are no less than the *minimum support* threshold. If the support of an itemset is less than the minimum support, then this itemset is an *infrequent itemset*.

In order to find out the frequent itemsets from a transaction database, there are many algorithms proposed by the previous studies [1–6]. However, in the real-world, the transactions will increase with time, and users may want to catch this real-time information to make advantageous decisions. Therefore, how to find the frequent patterns in time when a set of new transactions comes with time becomes an important issue. The straightforward way is to combine the original dataset and the new dataset together and re-mine the whole dataset. It is very inefficient to re-mine the whole updated database, since the database may continuously grow up and become very huge. Therefore, some researches [7–12] try to make use of the previous information without re-mine the updated database. There are two kinds of main algorithms which are divided by their used structures: One is the Apriori-like algorithm and the other is the Tree-based algorithm.

Because the Apriori-like algorithms [7, 8] need to take a lot of time to generate candidates, scan the database many times to search for the large number of candidates, such that there are few researches which use the Apriori-like structure in recent years. On the other hand, the Tree-based algorithms [9, 10, 11, 12] merge the new incoming data into the tree structure which was built to store the original dataset. In order to keep the information of frequent itemsets in the tree, they need to take a lot of time to insert, delete or move and combine the sub-tree structures. Therefore, Leung et al. [12] purposed a CAN Tree structure to store all the information of the transaction database in the tree. When the transactions are coming, it only needs to put the new transactions into the original tree structure

Table 1 The original transaction database

TID	Items
1	A, B, D
2	B, C
3	B, C, D, E
4	C

directly without any change for the tree structure. However, it not only wastes a large amount of memory space to store infrequent items, but also slows down the mining speed because of many unuseful information stored in the tree structure.

In order to avoid generating candidates, storing all the information of the transaction database, taking a lot of time to change the tree structure, and re-scanning the original database, we propose two efficient algorithms IMFP (Incrementally Mining Frequent Patterns) and IMFPC (Incrementally Mining Frequent Patterns with Complement) to find all the frequent patterns when a set of new transactions is added. Our storage structure is based on FP-tree structure [3] and there is a TID Set or its Complement Set recorded on each node of the tree structure, which records the TIDs of the transactions containing the items in the path from the node up to the child of the root. Our algorithms can easily determine which frequent items turn out to be infrequent and which infrequent items turn out to be frequent when a set of transactions is added. After removing the nodes which contain the infrequent items, the tree structure can be updated by simply performing set difference or intersection operations without scanning the original database.

2 Related Work

The early approaches for mining frequent itemsets are based on Apriori-like approach [1], which iteratively generate candidate $(k + 1)$-itemsets from the frequent k-itemsets $(k \geqq 1)$ and check if these candidate itemsets are frequent. However, in the cases of extremely large input sets or low minimum support threshold, the Apriori-like algorithms may suffer from two main problems of repeatedly scanning the database and searching for a large number of candidate itemsets.

In order to avoid generating a large number of candidate itemsets and scanning the transaction database repeatedly to count supports for the candidate itemsets, Han et al. [3] proposed an efficient algorithm *FP-Growth*. This algorithm constructs a frequent pattern tree structure which is called *FP-tree*. FP-tree consists of a null root, a set of nodes and a *header table*. Each node, except the root node, in the FP-tree consists of three fields: *item-name*, *count*, and *item-link*. The item-name registers which item this node represents, count registers the number of transactions represented by the portion of the path reaching this node, and item-link links to the next node in the FP-tree carrying the same item or null if there is none. There is an item-link structure for each frequent item. Each entry in the header table consists of two fields: *item-name* and *head of item-link* which points to the first node in the FP-tree carrying the same item-name.

The construction of FP-tree algorithm is described as follows: First, a null root node is created. For each transaction t in the database D, the frequent items in the transaction t are sorted by their supports in support descending order and the infrequent items in t are removed. Let n_0 is a root node. For the sorted transaction $\{i_1, i_2, \ldots, i_m\}$, the count of node n_j $(1 \leq j \leq q)$ of path $P = < n_0, n_1, n_2, \ldots n_q, \ldots n_r >$ $(r \geq 1)$ in the FP-tree adds 1 and a new node with item i_{k+1} and count 1 is created as a child of the node with item i_k $(\forall\ k, q \leq k \leq m\text{-}1)$ if the item of node n_i is t_i $(\forall\ i, 1 \leq i \leq q)$ and the item of node n_{q+1} is not i_{q+1}. FP-Growth algorithm requires only two full I/O database scans to build an FP-tree in main memory and then recursively mines frequent patterns from this structure by building *conditional FP-trees*.

For mining frequent itemsets from incremental databases, Cheung et al. designed the FELINE (FrEquent/Large patterns mINing with CATS trEe) algorithm [9] with a CATS Tree (Compressed and Arranged Transaction Sequences Tree). CATS Tree contains all the transactions in the transaction database. After scanning the database once, a CATS Tree can be constructed. When a new transaction is added, it is added at the root level. The items of the new transaction are compared with the items contained in the child node of the root or its descendant node at each level. If they are the same, then the items are merged with the nodes and the frequency counts of these nodes are incremented by 1. Otherwise, a new branch is created which is formed by the remaining items of the transaction, and the frequency counts of the nodes contained the remaining items are set to 1. If the support of an item in a node becomes larger than that of its ancestor, the two nodes need to be swapped, since items are arranged in a descending local frequency order in the CATS Tree. Therefore, the mining process needs to traverse both upwards and downwards of the CATS Tree to include all the frequent items. The FELINE algorithm suffers from the problems described as follows. First, though CATS Tree is constructed as compact as possible, it is not guaranteed to have the highest compression. Second, the mining process requires extra cost for the swapping and merging of the nodes. Third, for a node, not only its ancestors need to be traversed, but also its descendants, such that the mining process is inefficient.

Leung et al. proposed CAN Tree [12] (Canonical-order Tree) in 2005. CAN Tree also contains all the transactions in a transaction database, and therefore it needs to scan the database only once to construct a CAN Tree. The items in the nodes of the CAN Tree are arranged according to a canonical order, which can be arranged in lexicographic or alphabetical order. Notice that, once the ordering is determined for the original database, items will follow this ordering in a CAN Tree for subsequently updated databases. Finally, the frequent itemsets can be found by applying FP-Growth algorithm [3]. Because all the items including infrequent items are stored in a CAN Tree and the items are not ordered in their frequency order, it needs to spend a large amount of memory space to store the large CAN Tree, and take a lot of time for mining frequent itemsets from the large CAN Tree.

In 2008, Tzung-Pei et al. proposed FUFP (Fast Updated FP-tree) algorithm [10]. The FUFP Tree construction algorithm is the same as the construction of an FP Tree before new transactions coming. When the new transactions are added, The items are partitioned into the following four cases according to whether they are frequent or infrequent in the original database and in the updated database: (1) If the items are frequent both in the original database and in the updated database, there is no change for the nodes containing these items in the FUFP Tree, since these items are frequent before and after adding the new transactions. (2) If an originally frequent item becomes infrequent, then the nodes containing this item are directly removed from the FUFP Tree. (3) If an originally infrequent item becomes frequent, then a new node containing the item is inserted into the leaf of the FUFP Tree. (4) If the items are infrequent both in the original database and the updated database, then there is nothing to do. After updating the FUFP Tree, the frequent itemsets can be found by applying FP-Growth algorithm [3]. However, FUFP algorithm still suffers from some problems: FUFP Tree is not compact, since the new frequent items are always added to the leaf nodes of the tree, such that the tree may become larger and larger. Besides, when an originally infrequent item becomes frequent, it is necessary to re-scan the original database to obtain the support for the item in the updated database.

3 Algorithm IMFP

The structure of FP Split [4] is used in our storage structure. First, we scan the original transaction database once to count the support for each item and keep the set of the transaction identifiers (TIDs), which contain this item. Table 2 shows the TID Set for each item in the original database (Table 1). We also can easily get the support of each item by counting the number of elements in its TID Set. Assume that the user-specified minimum support threshold is 50 %. That is, the minimum support count is 2. The frequent items are ordered by their support counts in descending order. In the example, the ordered frequent items are B, C, and D.

In the following, we use the example to illustrate how to construst the tree structure [4]. First, A null root and a header table is created, which is the same as the header table in the FP-Tree [3]. Each node, except the root node, on the tree includes an item, the frequency count of the item in the path and a TID Set. The frequent items are added to the tree according to their orders. In the following, for a node which contains item X, we call it node X. Since item B is in the first order, a child node of the root node, which contains item B, is created, and its TID Set is {1, 2, 3}, which is shown in Fig. 1. The corresponding item-link of item B in the header table is also linked to the node B.

When an item is ready to be added to the tree, the nodes in the tree need to be visited beginning from the left child of the root, and there are two required steps to be followed: (1) Identify if it is necessary to move down to the children of the visited

Table 2 The TID Set for each item in Table 1

Items	TID
A	1
B	1, 2, 3
C	2, 3, 4
D	1, 3
E	3

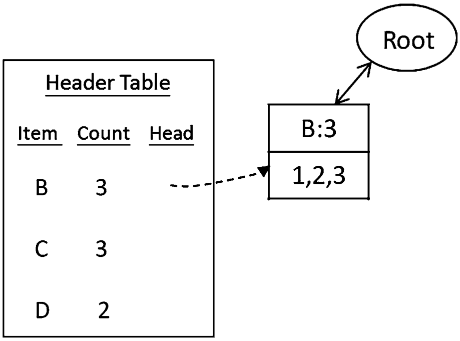


Fig. 1 The tree structure after adding item B

node. We need to do the set intersection on the two TID Sets of the added item and the visited node. In this example, the next item to be added is item C, and the first visited node in the tree is node B. Our algorithm performs the set intersection on the two TID Sets of the added item C and the visited node B. The result is {2, 3}, which indicates that both item B and item C appear in the two transactions with identifiers 2 and 3 at the same time. Therefore, a new node C is created as the child node of node B, and its TID Set is {2, 3}, which is shown in Fig. 2. (2) Identify if it is necessary to move on to the siblings of the visited node. We need to do the set difference on the TID Sets of the added item and the visited node. For the above example, after identifying if it is necessary to move on to the siblings of the visited node, we do the set difference on the TID Set {2, 3, 4} of the added item C and the TID Set {1, 2, 3} of the visited node B. The result is the TID Set {4}, which means that item C is contained in the transaction with TID 4, but item B is not. As a result, we need to create a new node which contains item C as a sibling of node B, and its TID Set is {4}, which is shown in Fig. 3.

For each frequent item, it can be built on the tree through the above two steps whenever they meet a node. After finishing adding item C, the next item to be added is item D. For the current tree structure, the first visited node is node B. Our algorithm performs the set intersection on the two TID Sets of item D and the visited node B. The result of the intersection is {1, 3}, which implies that item D and item B are both contained in the transactions with TID 1 and TID 3. As a consequence, we can move down to the children of node B, and node C is visited. Besides, the set difference on the two TID Sets of item D and node B is also performed. The result is empty, which indicates that all the transactions containing item B also contain item D. Thus, it is unnecessary to move on to the siblings of node B.

Fig. 2 The tree structure after adding C as a child of node B

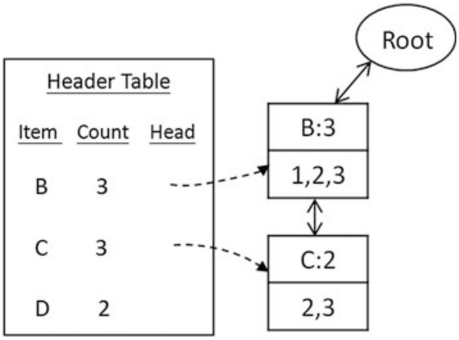
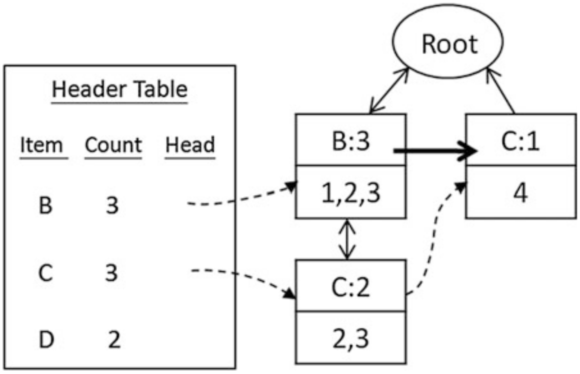


Fig. 3 The tree structure after adding C as a sibling of node B



After that, the next visited node is the child node C of node B. Similarly, the set intersection on the two TID Sets of item D and the visited node C is performed and the result is {3}, which means that items B, C, and D are contained in the transaction with TID 3. We can move down to the child of the node C. Because there is no any child for node C, a new node containing item D is created as a child of the node C, and its TID Set is {3}, which is shown in Fig. 4.

After finishing the intersection, we keep doing the set difference on the two TID-ists of item D and the current visited node C which is the child of node B. The result of the difference is {1}, which implies that item D is contained in the transaction with TID 1 but item C is not. Therefore, a new node containing item D is created as a sibling of node C, which is shown in Fig. 5. Since all the frequent items have been added to the tree, the tree construction is finished. The frequent itemsets can be found by applying FP-Growth algorithm [3].

When a set of the new transactions are added, our algorithms first scan the new transaction database once to obtain the TID Sets of each item and its support count in the new transaction database. In the following, the original database is denoted as DB, and the new transaction database is denoted as db. When a set of the transactions are added, each item in the new transaction database will meet one of the following three cases: (1) The item is frequent in DB, but becomes infrequent

Fig. 4 The tree structure after adding D as a child of node C

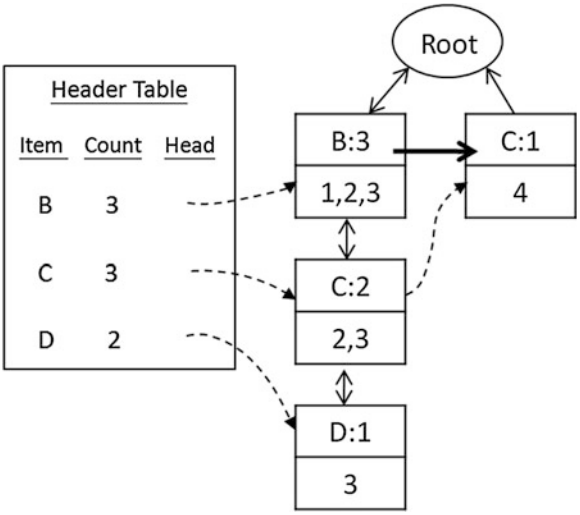
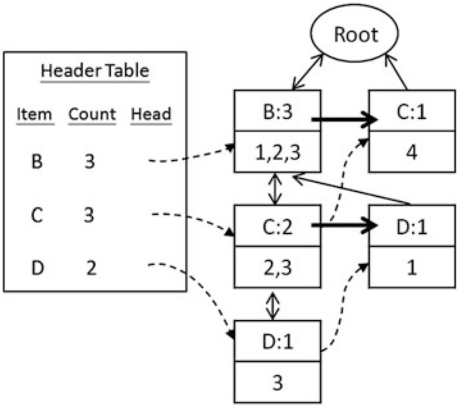


Fig. 5 The tree structure after adding D as a sibling of node C



in the updated database DB ∪ db. (2) The item is frequent both in DB and DB ∪ db. (3) The item is infrequent in DB, but becomes frequent in DB ∪ db. Our algorithm sequentially processes the items from the 1st case to the 3rd case.

In the following, we use an example to explain how to update the tree structure and find all the frequent itemsets when a set of transactions is added for our algorithm IMFP. Table 3 shows the added transactions which are added to Table 1. The two new transactions in Table 3 are scanned to get the TID Set and the support count for each item. The TID Set for each item in Table 3 is shown in Table 4.

In this example, the support counts for item D in the original database and in the set of the added transactions are 2 and 0, respectively. Therefore, the support count for item D in the updated database DB∪db is 2, and the minimum support count for the updated database(Tables 1 and 3) is 3. Item D is frequent in the original

Table 3 The set of the added transactions

TID	Items
5	B, A
6	A, E, C

Table 4 The TID Set for each item in Table 3

Items	TID
A	5, 6
B	5
C	6
E	6

database DB but infrequent in the updated database, which meets the 1st case. In this case, all the nodes with item D are marked and removed from the tree structure. Because items B, C are frequent in DB, and items B and C remain frequent in DB ∪ db, items B and C meet the 2nd case.

For the items in the 1st case, since there are some nodes which contain the items on the original tree structure, these nodes should be removed after adding the transactions. If the removed node X is a leaf node, it can be removed directly. Otherwise, if there are a child node and a sibling node of X which contain the same item, then the two nodes need to be merged, that is, to combine their TID Sets and sum their counts. If there is no any sibling node of X with the same item as a child node Y of X, then the parent node of X becomes the parent of Y. For the above example, because D is in the 1st case and the nodes with item D are both leaf nodes, we only remove the nodes without merging, which is shown in Fig. 6.

After removing the items in the first case from the tree, our algorithm processes the items in the 2nd case. Since the items have existed in the tree before adding the transactions and they are remained in the tree after adding the transactions, we only need to update the counts and TID Sets on the nodes with the items. For the above example, the support counts of both items B and C in the original database are 3, and in the set of the added transactions are 1. Therefore, the support counts of both items B and C in the updated database are $3 + 1 = 4$. Therefore, items B and C are frequent both in the original database and in the updated database. In this case, items B and C are already in the tree structure. Our algorithm IMFP processes these items according to their orders and uses the similar method of the tree construction to update and add these items to the tree, but only consider the new TID Sets (e.g., Table 3).

Since the item order is item B and then item C, the frequency count and TID Set for the child node B of the root node in Fig. 6 are updated as 4 and {1, 2, 3, (5)}, respectively. The next item to be processed is item C. IMFP visits each node in the tree from the left child of the root. The first visited node is node B. IMFP performs the set intersection on the new TID Set {6} of the updated item C and the new TID

Fig. 6 The tree structure after removing the nodes with D

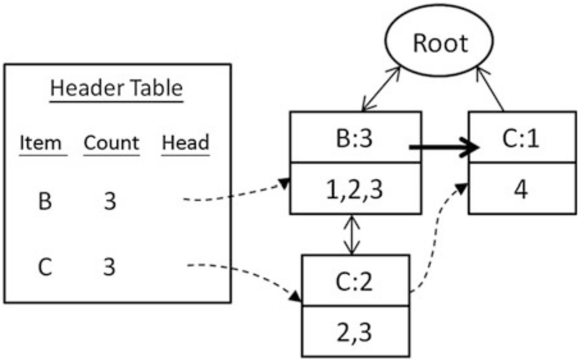
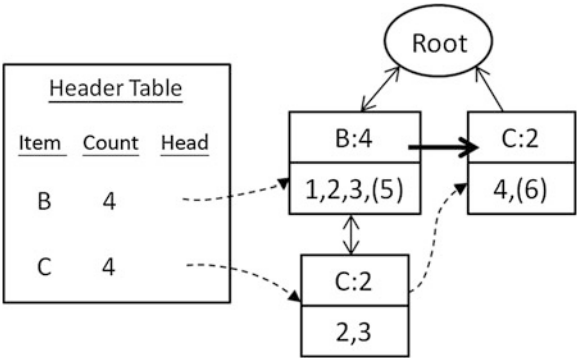


Fig. 7 The tree structure after processing item C



Set {5} of the visited node B, which is empty. That is, it is not necessary to move down to the children of the visited node.

After that, we do the set difference on the new TID Set {6} of the item C and the new TID Set {5} of the visited node B. The result is {6}, which means that item C is contained in the added transaction but item B is not. As a result, the frequency count and TID Set of node C, which is a sibling of node B, are updated as 2 and {4, (6)}, respectively, which is shown in Fig. 7.

The support counts for item A in the original database and in the set of the added transactions are 1 and 2, respectively. Therefore, the support count for item A in the updated database is 3. Item A is infrequent in the original database but frequent in the updated database. Therefore, item A with TID Set {1, 5, 6} is added into the leaf nodes of the tree. In this case, the method to add an item into the tree is the same as the method for the tree construction. After adding item A to the current tree structure (i.e., Fig. 7), the final tree structure is shown in Fig. 8. For the item E which is infrequent in both of the original database and the updated database, we only need to update its support count and TID Set without changing the tree structure.

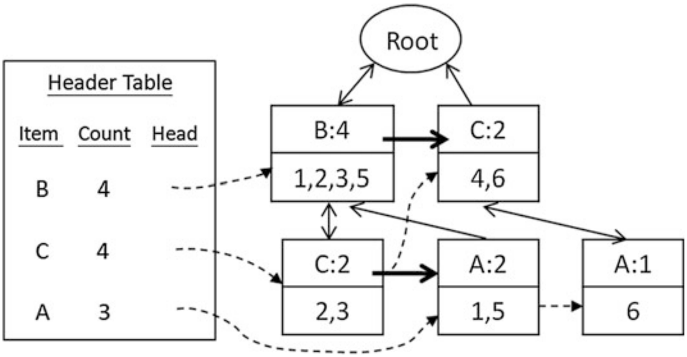


Fig. 8 The tree structure after adding item A

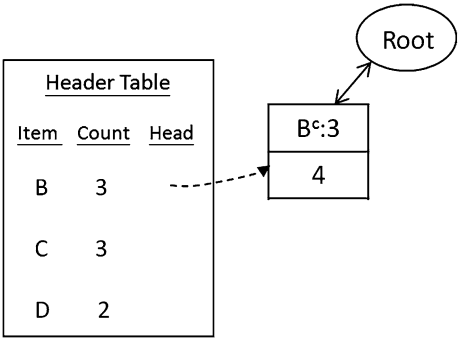
Although IMFP can only consider the added transactions, and easily perform set intersection and set difference to update the tree structure when the transaction database is growing, it needs to store TIDs in each node in the tree structure, such that a large number of memory space need to be used to store the large tree structure. In order to reduce the memory usages, we propose another algorithm IMFPC to reduce the number of the elements in the TID Sets.

4 Algorithm IMFPC

In this section, we describe how to reduce the memory space used by the tree structure and how to update the tree structure when the new transactions are added to the database. In the following, we also use the above example to illustrate how to construct the tree structure for our IMFPC algorithm from Table 1.

First, there is only an empty root node created on the tree and a header table whose structure is the same as the header table in the FP-tree [3]. According to the item order, item B and its TID Set are added into the tree. Because there is no other node but the root on the tree, a node with item B is created as the child node of the root and the TID Set is recorded on the node. In order to reduce the memory usage, the TID Set for a node will be transformed into its *Complement Set* if the count for the node is greater than half of the total number of the transactions in the database. We call a node with a TID Set as a T-node, and a node X with a Complement Set as a C-Node which is denoted as X^C. Since the count of node B whose TID Set is {1, 2, 3} which is greater than 4/2 = 2, the TID Set of node B is transformed to its complement set {4}. Therefore, node B is a C-node which is denoted as B^C. The corresponding item-link of item B in the header table is also linked to the node B^C. After adding item B and its TID Set, the tree structure is shown in Fig. 9.

Fig. 9 The tree structure after adding item B



When the next item Y is added into the tree, the tree structureis traversed in depth-first order, and there are two steps to perform when a node X is visited:

(1) To check whether X's child node need to be visited, that is to do the "*Downward Operation*" on item Y and node X to get the common occurrences of items X and Y in the same transactions. The operation can be divided into the following cases: If X is a T-Node, we perform the intersection on the two TID Sets of item Y and node X, that is T(Y) ∩ T(X); if X is a C-Node, we perform the set difference on the TID Set of item Y and the Complement Set of node X, that is T(Y) − C(X). If the result is not empty, then our algorithm check if there is a child node of node X. If node X has a child node Z, then the downward and upward operations continue to perform on item Y with the result set and the node Z. Otherwise, a node with item Y and the result set is created as a child node of X. For the above example, the next item which needs to be processed is item C. The downward operation on item C and the visited node B^C is performed, that is {2, 3, 4} − {4} = {2, 3}, which represents that items B and C have common occurrences in the transactions with TIDs 2 and 3. Therefore, a node C with TID Set {2, 3} is created as node B^C's child node, which is shown in Fig. 10. Since the count of the child node C is no greater than 2, the TID Set does not need to be transformed.

(2) To check whether X's sibling node needs to be visited, that is to do the "*Rightward Operation*" on item Y and node X to get the set of the TIDs of the transactions which contain item Y but no item X. The operation also can be divided into the following cases: If X is a T-Node, we perform the set difference on the two TID Sets of item Y and node X, that is T(Y) − T(X). If X is a C-Node, we perform the intersection on the Complement Set of node X and the TID Set of item Y, that is C(X) ∩ T(Y). If the result is not empty, then our algorithm check if there is a sibling node of X. If node X has a sibling node Z, then the downward and upward operations continue to perform on item Y with the result set and the node Z. Otherwise, a node with item Y is created as a sibling node of X.

For the above example, the rightward operation on item C and node B^C is performed, that is {2, 3, 4} ∩ {4} = {4}, which means that the transaction TID 4 contains item C but no item B. Therefore, a node C with TID Set {4} is created as

Fig. 10 The tree structure after adding C as a child of node B^C

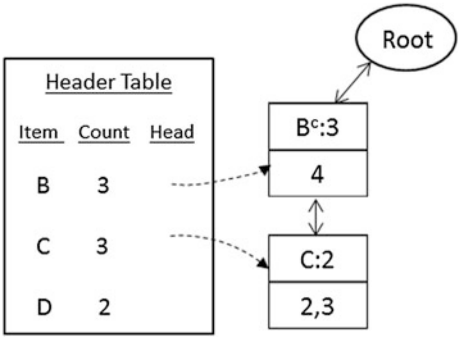
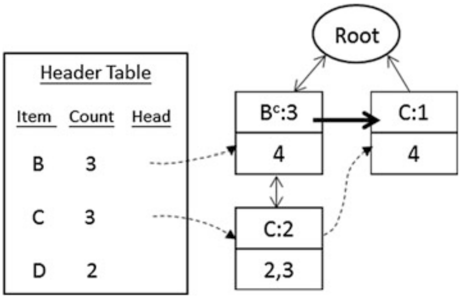


Fig. 11 The tree structure after adding C as a sibling of node B^C



the sibling node of node B^C, which is shown in Fig. 11. After that, whenever a new incoming item visits a node on the tree, the two operations need to be performed to determine if a new node need to be created. Figures 12 and 13 show the tree structure after performing the downward and rightward operations on item D and the visited nodes, respectively. After processing all the frequent items, the tree structure corresponding to the original database (Table 1) is constructed.

When a set of the new transactions is added, our algorithm first scans the new transaction database once to obtain the TID Set of each item and its support count in the new transaction database. In the following, we use the above example to explain our algorithm IMFPC. In this example, Table 1 is the original database and Table 3 is the new transaction database. Because items B, C and D are frequent in DB, and items B and C remain frequent in DB∪db, items B and C meet the 2nd case, and item D meets the 1st case. The method to remove the items in the 1st case from the tree structure for IMFPC is the same as that of algorithm IMFP. For this example, because D is in the 1st case and the nodes with item D are both leaf nodes, we only remove the nodes without merging, which is shown in Fig. 14.

After removing the item D from the tree structure, our algorithm processes the items in the 2nd case. Since the items have existed in the tree before adding the transactions and they are remained in the tree after adding the transactions, we only need to update the counts and TID Set on the nodes with the items. If the count for a node is greater than half of the number of the transactions in DB∪db,

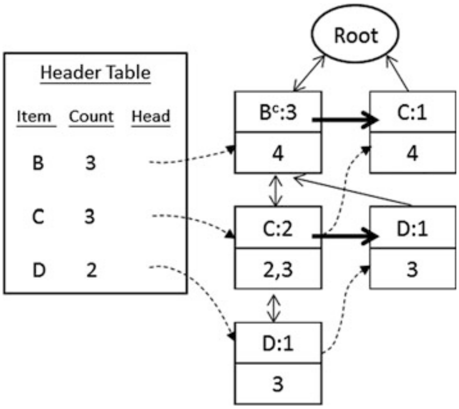


Fig. 12 The tree structure after adding D as a child of node C

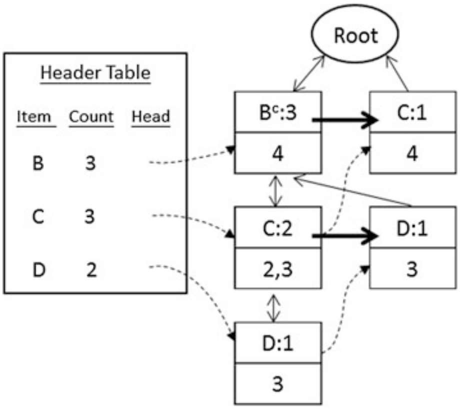


Fig. 13 The tree structure after adding D as a sibling of node C

the TID Set will be transformed to its complement set. For the above example, the nodes with item B or item C should be updated. According to Table 4, the item B with TID 5 is first added into the tree structure. The first visited node in Fig. 14 is node B^C which contains the same item B. Therefore, the TID 5 is added into the TID Set of node B. Since the count of the node B is 4 which is greater than 6/2 = 3 after adding the transactions, it remains to record the complement set of the TID Set, that is {4, 6}, which is shown in Fig. 15. To avoid unnecessary computation on the original TID Sets, the new TIDs are stored individually and combined with the original TID Sets after processing all the items in the 2nd case. Therefore, the operations only need to be performed on the new TIDs.

Another item in the 2nd case is item C which appears in the transaction TID 6 in db. The node B^C in Fig. 15 is first visited. Our algorithm performs the downward operation on the node B^C and item C with TID Set {6}, which is $\{6\} - \{6\} = \varphi$. Therefore, the child node of node B^C does not need to be visited. Since the rightward operation on the item C and the visited node B^C is $\{6\} \cap \{6\} = \{6\}$, the

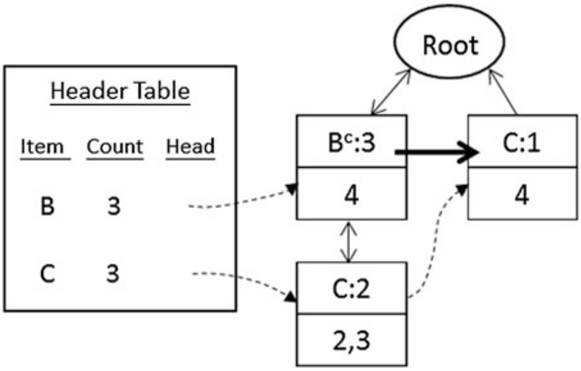


Fig. 14 The tree structure after removing the nodes with D

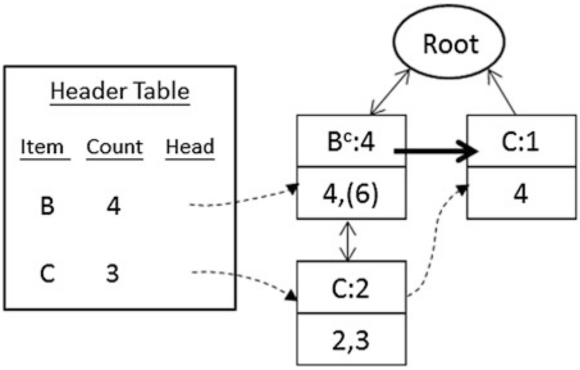


Fig. 15 The tree structure after processing item B

sibling node of node B^C need to be visited and the visited node is node C. Therefore, the TID 6 is directly added into the TID Set of node C, which becomes {4, 6}. After processing all the items in the 2nd case, the new TID Sets can be really combined with the original TID Sets, which is shown in Fig. 16.

For the items in the 3rd case, since the items are not frequent in the original database, but become frequent after adding the new transactions, these items need to be added into the tree after adding the new transactions, which is the same as the method to add the items tree structure in the tree construction phase. For the example, item A is in the 3rd case. Because there are no nodes with item A on the tree, the new transactions TID 6 which contains item A needs to be combined with the TID Set of item A in the original database, that is, the new TID Set of item A is {1, 5, 6}. Therefore, item A with TID Set {1, 5, 6} is added into the tree: The node B^C is first visited. The downward and rightward operations are performed on item A with TID Set {1, 5, 6} and node B^C, which are $\{1, 5, 6\} - \{4, 6\} = \{1, 5\}$ and $\{1, 5, 6\} \cap \{4, 6\} = \{6\}$, respectively. Therefore, the child node of node B^C needs to be visited and the visited node is node C. After performing the downward and rightward operations on item A with TID Set {1, 5} and the node C, that is $\{1, 5\} \cap \{2, 3\} = \varphi$ and $\{1, 5\} - \{2, 3\} = \{1, 5\}$, respectively, the child node of node C

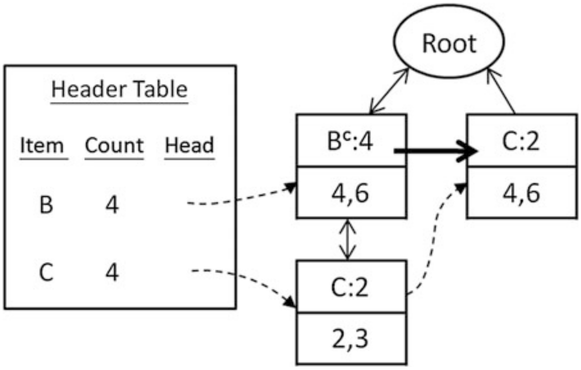


Fig. 16 The tree structure after combining the original and new TID Sets

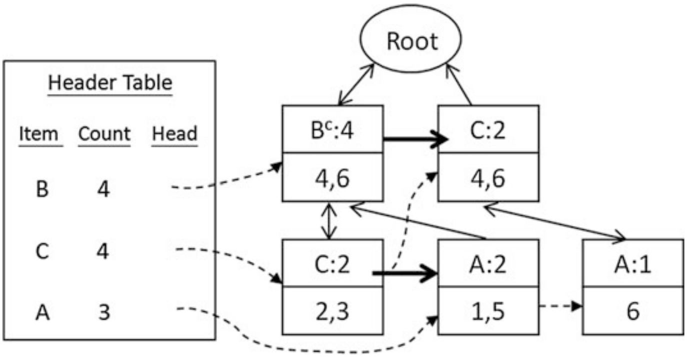


Fig. 17 The tree structure after adding item A

does not need to be visited, but the sibling node of node C needs to be visited. Therefore, a new node A with TID Set {1, 5} is created as the sibling node of node C. IMFPC goes back to node B^C and visits the sibling node C of the node B^C. After performing the downward and rightward operations on item A with TID Set {6} and the node C, that is $\{6\} \cap \{4, 6\} = \{6\}$ and $\{6\} - \{4, 6\} = \varphi$, respectively, a new node A with TID Set {6} is created as a child node of the node C, since node C has no any child node, which is shown in Fig. 17.

5 Experimental Results

In this section, we evaluate the performance of our algorithm and compare it with the algorithm FUFP [10] in the same execution environment. The experiments are performed in Java on a computer equipped with Intel® Core™ i5 Quad CPU 760

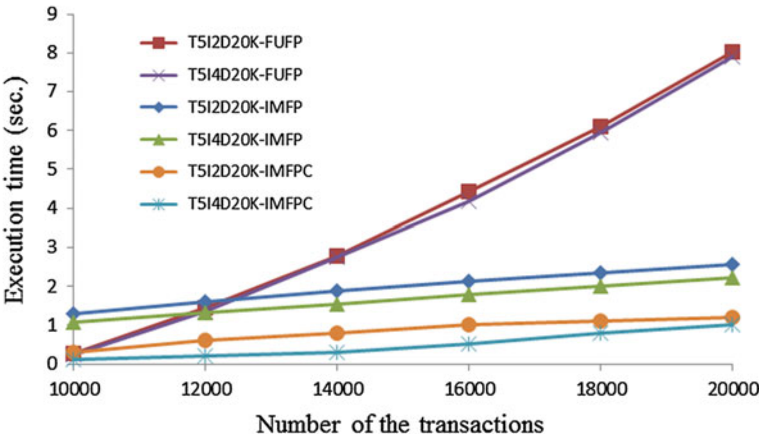


Fig. 18 Execution times for the three algorithms on datasets T5I2D20 and T5I4D20K

@ 2.80 GHz and 2 GB main memory and running on the Microsoft Windows 7 operating system. We generate eight synthetic datasets T5I2D20K, T5I4D20K, T10I2D20K, T10I4D20K, T20I2D20K, T20I4D20K, T10I2D100K and T10I4D100K by using IBM Synthetic Data Generator [13], in which T is the average length of the transactions, I is the average size of maximal potential frequent itemsets and D is the total number of transactions. The number of distinct items is set to 100K.

Because the tree structures of our algorithm and FUFP algorithm are the same, and both of them use FP-Growth algorithm for mining frequent itemsets, the mining times for the two algorithms are the same, too. Therefore, we only evaluate the performance of the tree construction and updating for our proposed two algorithms and FUFP algorithm.

We take 10K transactions from the three sets of the datasets T5I2D20K and T5I4D20K, T10I2D20K and T10I4D20K, T20I2D20K and T20I4D20K as the original databases, and accumulate the execution times for adding 2K transactions every time to these original databases. Figures 18, 19 and 20 show the execution times for the three algorithms on the three sets of the datasets when the minimum support is set to be 0.15 %. From these experiments, we can see that our algorithms outperforms FUFP on all the datasets and the performance gap increases as the number of the transactions in the updated databases increase, since FUFP needs to re-scan the original database to compute the supports for the itemsets which turn out to be frequent after adding the transactions, but our algorithms can obtain the supports for these itemsets from the TidSets without scanning the original database. However, in the beginning, our algorithms take more time to construct the original tree structure, since our algorithms need to take time to transform the original dataset into the TidList, but FUFP uses the same way as the FP-Growth algorithm [3] to construct a FP-tree. Besides, IMFPC slightly outperforms IMFP, since IMFPC stores the complement set of the TID Set if the

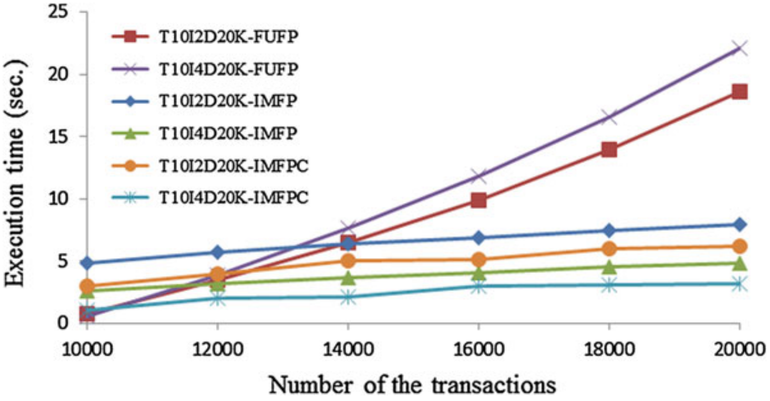


Fig. 19 Execution times for the three algorithms on datasets T10I2D20K and T10I4D20K

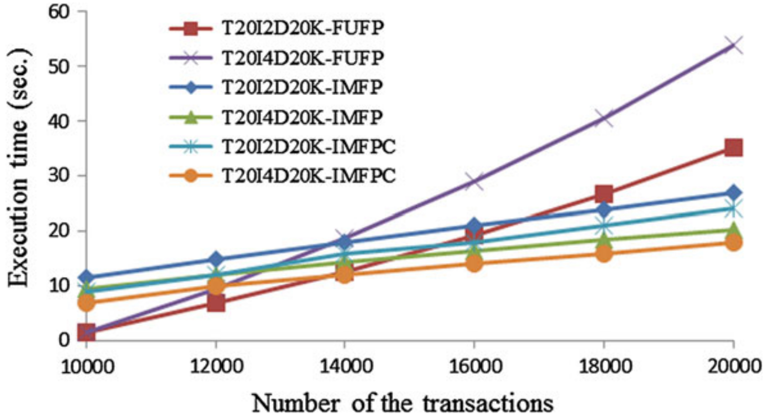


Fig. 20 Execution times for the three algorithms on datasets T20I2D20K and T20I4D20K

complement set is shorter than the TID Set. Therefore, IMFPC can perform the set operations on the shorter set than IMFP.

Figure 21 shows the execution times for FUFP and our algorithms on the larger datasets T10I2D100K and T10I4D100K. In this experiment, We take 50K transactions from the dataset as the original database and accumulate the execution times for adding 10K transactions every time to the original database. From Fig. 21, we can see that the execution times only slightly increase as the number of the transactions in the updated database increases for our algorithms. Our algorithms are more stable than FUFP and outperform FUFP when the transactions are continuously added into the original database. Figure 22 shows the memory usages for the three algorithms on dataset T10I4D100K, from which we can see that FUFP uses less memory space than our algorithms, since our algorithms need to store the TID Set or its complement set on each node of the tree structure, but

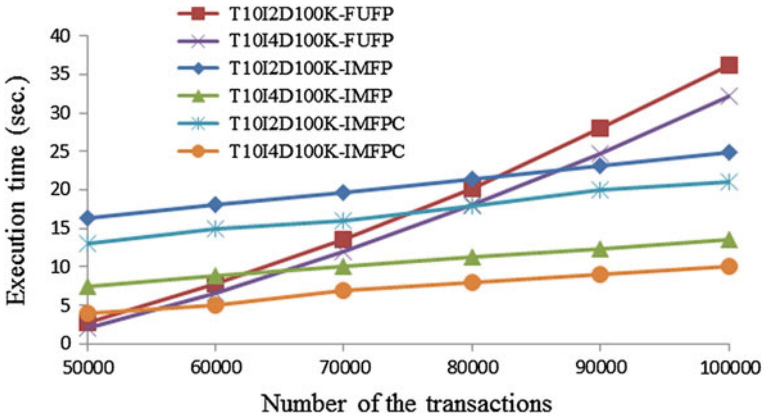


Fig. 21 Execution times for the three algorithms on datasets T10I2D100K and T10I4D100K

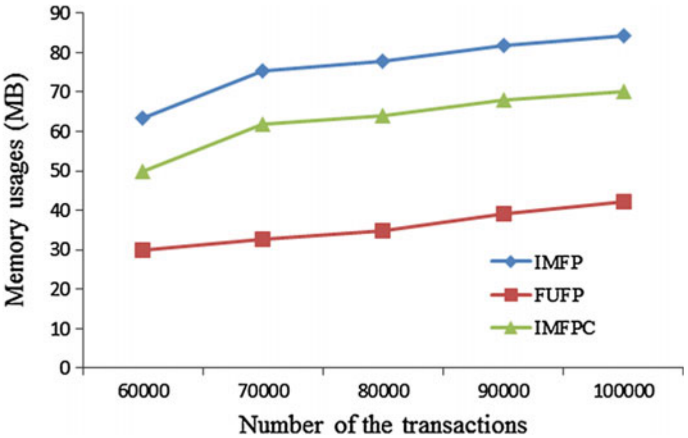


Fig. 22 Memory usages for the three algorithms on dataset T10I4D100K

FUFP does not need to store this information. IMFPC uses less memory space than IMFP, since IMFPC stores the complement set of the TID Set if the elements of the complement set are less than that of the TID Set, that is, IMFPC stores less TIDs on each node than IMFP.

6 Conclusions

This chapter proposes two efficient algorithms IMFP and IMFPC for incrementally mining frequent patterns. When a set of the transactions is added, our algorithms only need to apply the set difference or intersection operations on the TID Set or

Complement Set, and adjust few tree nodes without re-scanning the original database. Therefore, our algorithms outperform the previous approaches. In order to reduce the memory usages for IMFP, IMFPC stores the complement set instead of TID Set on a node of the tree structure if the number of the elements in the TID Set is more than that of its complement set. Therefore, IMFPC can take less time to perform the set operations and less memory space to store the TIDs than that of IMFP.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), pp. 487–499 (1994)
2. Mohammad, E.H., Osmar, R.Z.: COFI-tree mining: a new approach to pattern growth with reduced candidacy generation. In: Workshop on Frequent Itemset Mining Implementations (FIMI'03) in conjunction with IEEE-ICDM
3. Jiawei Han, Jian Pei and Yiwen Yin, "Mining Frequent Patterns without Candidate Generation. In: Proceedings of the 2000 ACM International Conference on Management of Data (SIGMOD), pp. 1–12 (2000)
4. Lee, C.-F., Shen, T.-H.: An FP-split method for fast association rules mining. In: Proceedings of the 3rd International Conference on Information Technology: Research and Education (ITRE), pp. 459–463 (2005)
5. Wang, K., Tang, L., Han, J., Liu, J.: Top down FP-growth for association rule mining. In: Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, pp. 334–340 (2002)
6. Yen S.J., Lee, Y.S., Wang C.K., Wu C.W.: The Studies of Mining Frequent Patterns Based on Frequent Pattern Tree. *PAKDD 2009*, LNAI 5476, pp. 232–241 (2009)
7. Cheung, D.W., Han, J., Ng, V.T., Wong, C.Y.: Maintenance of discovered association rules in large databases: an incremental updating technique. In: Proceedings of the 12th International Conference on Data Engineering (ICDE), pp. 106–114 (1996)
8. Cheung, D.W., Lee, S.D., Kao, B.: A general incremental technique for maintaining discovered association rules. In: Proceedings of the 5th International Conference on Database Systems for Advanced Applications (DASFAA), pp. 185–194 (1997)
9. Cheung, W., Zaïane, O.R.: Incremental mining of frequent patterns without candidate generation or support constraint. In: Proceedings of the 7th International Database Engineering and Applications Symposium (IDEAS), pp. 111–116 (2003)
10. Hong, T.-P., Lin, C.-W., Yu-Lung, W.: Incrementally fast updated frequent pattern trees. Expert Syst. Appl. Int. J. **34**(4), 2424–2435 (2008)
11. Koh, J.-L., Shieh, S.-F.: An efficient approach for maintaining association rules based on adjusting FP-tree structures. In: Proceedings of the 12th International Conference on Database Systems for Advanced Applications (DASFAA), pp. 417–424 (2004)
12. Leung, C.K.-S., Khan, Q.I., Hoque, T.: CanTree: a tree structure for efficient incremental mining of frequent patterns. In: Proceedings of the 5th IEEE International Conference on Data Mining (ICDM), pp. 274–281 (2005)
13. IBM Synthetic Data Generator http://www.almaden.ibm.com/software/quest/Resorces/index.shtml

Improved Latent Semantic Indexing-Based Data Mining Methods and an Application to Big Data Analysis of CRM

Jianxiong Yang and Junzo Watada

Abstract The rapid growth of services industry over these years has led to an increased number of research works in the improvement of service quality by data mining. However, analyzing service quality and determining the factors in influencing consumer's perception of service quality are a challenging issue. In this paper, we introduce some data mining methods from a basic one to an advance one. Finally, we use these methods to resolve Customer Relationship Management (CRM) cases and compare their efficiency. We apply statistical and machine learning techniques to study the dynamic customer level between the occurrence frequencies of events in users' feedback and the corresponding Customer Satisfaction Index (CSI). Based on our analysis we observed that in the context of customer support centers, service experience has strongly influence on perceived customer satisfaction and service quality. Based on the research results an improved approach for innovative CRM is presented. The thesis proposes three methods and explains an application to big data analysis for CRM at the end.

Keywords Web mining · Rough set · Attribute reduction matrix (ARM) · Singular value decomposition (SVD) · Latent semantic indexing (LSI) · Customer relationship management (CRM)

J. Yang · J. Watada (✉)
Graduate School of Information, Production and Systems, Waseda University, 2-7,
Hibikino, Wakamatsu, Kitakyushu 808-0135, Japan
e-mail: watada@waseda.jp; junzo.watada@gmail.com

J. Yang
e-mail: leoworldplus@yahoo.co.jp

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_7,

1 Introduction

In recent years Knowledge representation and manipulation are required in a number of artificial intelligence algorithms. Knowledge representation has been the subject of intense study in the artificial intelligence community for some time. The goal of knowledge representation is to allow AI programs to behave as if humans do in solving. The schemes of knowledge representation include logic programming, semantic networks, procedural interpretation, production systems and frames. The use of knowledge representations can be found in automated deduction systems, inference machines, expert systems and knowledge bases.

In China, there are about 3,000 gas stations of every province. Every gas station serves about 800–1,000 cars on the average every day(24 h). Some of these customers are our faithful customers, some of them are general ones and some of them are new customers or faithful customers of other gas stations. If we want to obtain the biggest benefit, we must firmly catch the heart of faithful customers. How to firmly catch the heart of faithful customers is the object of this paper. Member card will record basic information of customer which includes Car No, consumption of add gas, and buy some water, auto parts and etc. By their quantity and frequency stability of adding gas, consumption and frequency stability of shop goods of gas station, we can use clustering and Latent Semantic Index (LSI) method to analyze and cluster who the faithful customers are.

The clustering methods can be classified into two groups [1]: hierarchical method and partitioning method. Partitioning clustering can be classified into hard clustering and overlapping clustering (e.g., fuzzy clustering). Any given document has a possibility to contain multiple subjects or categories. This issue aims to use a fuzzy clustering approach, which enables us to include a document in multiple clusters. The method is different from hard clustering method, which a document only belongs to one cluster, not more. The hard clustering assumes well defined boundary among the clusters.

While grouping or clustering of documents, the problem is a very huge number of terms or words are contained in the full text vector space [2]. Many lexical matching at term grade are inaccurate. Sometimes, a word has ambiguously several meaning or some words have synonymously the same meaning, and these cause results in the selection of irrelevant documents. Therefore, this research uses an Information Retrieval approach of LSI. In this method, the documents are projected onto a small subspace of this vector space and are clustered. So, there is creation of new abstract vector space is capable of capturing contents of important documents in order [3].

The thesis consists of the following structure. Chapter “Nearest Neighbor Queries on Big Data” explains the background and survey of the research. Chapter “Information Mining for Big Information” gives a simple fuzzy clustering technique. Chapter “Information Granules Problem: An Efficient Solution of Real-Time Fuzzy Regression Analysis” explains Latent Semantic Indexing for Data Mining. Chapter “How to Understand Connections Based on Big Data: From

Cliques to Flexible Granules" explains Rough Set Based Optimization for Data Mining. Chapter "Graph-Based Framework for Evaluating the Feasibility of Transition to Maintainomics" concerns with the applications of data clustering for faithful customer analysis. Then, Chapter "Incrementally Mining Frequent Patterns from Large Database" draws the conclusions of these discussions.

2 The Fuzzy Clustering Methods

Fuzzy clustering is a class of algorithm for cluster analysis in which the allocation of data points to clusters. It is the process of dividing data elements into classes or clusters so that items in the same class are as similar as possible, and items in different classes are as dissimilar as possible. Depending on the nature of the data and the purpose for which clustering is being used, different measures of similarity may be used to place items into classes, where the similarity measure controls how the clusters are formed. Some examples of measures used in clustering are distance, connectivity, and intensity [4, 5].

2.1 Creating Intelligent Clustering System

2.1.1 Data Matrix

In the data records of data storage, data matrix can establish the classification data set, and quantitative target sets are analyzed. Let $\boldsymbol{M}$ be data set matrix, where $\boldsymbol{m}_{\mathrm{ij}}$ is an element of $\boldsymbol{k} \times \boldsymbol{n}$ array matrix $\boldsymbol{M}$, $\boldsymbol{j} = 1, 2, 3, \ldots,\ \mathrm{k}; \boldsymbol{i} = 1, 2, 3, \ldots,\ \mathrm{n}$.

2.1.2 Data Standardization

In the actual data, usually each different data set has a different dimension; therefore, we need to deal with the specified data standardization. It is necessary to normalize them to each cell between 0 and 1 in order that we can determine whether the value is big or small in a same measurement. Nevertheless, it is not possible to judge whether non-standardized value is big or small. After standardization, we can easily judge the biggest value in every set. In every set, "1" is the biggest common value. It like (Fig. 1):

(a) **Translational/transformation of standard deviation**

$$m'_{ij} = \frac{m_{ij} - \overline{m}_j}{S_j} \tag{1}$$

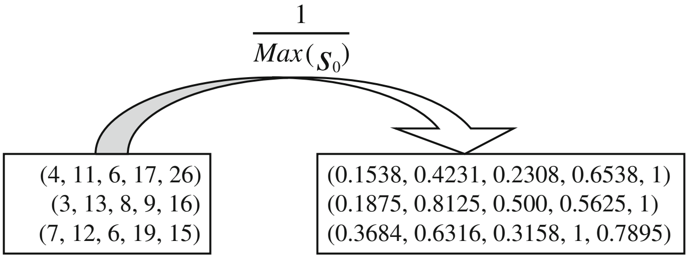


Fig. 1 The data standardization

where ***i*** is the number of column, ***j*** is the number of row, average $\overline{m}_j = \frac{m_{1,j}+m_{2,j}+m_{3,j}+\cdots+m_{kj}}{k}$, standardized deviation can be written as

$$S_j = \sqrt{\frac{1}{k}\sum_{j=1}^{k}(m_{ij} - \bar{m}_j)^2}$$

(b) Translational/transformation of [0, 1] range

After standardized deviation transformation, the m'_{ij} is uncertain in the interval [0, 1]. So it requires range transformation.

$$m''_{ij} = \frac{m'_{ij} - \min\limits_{1 \le i \le n}\{m'_{ij}\}}{\max\limits_{1 \le i \le n}\{m'_{ij}\} - \min\limits_{1 \le i \le n}\{m'_{ij}\}} \tag{2}$$

At last, the $\boldsymbol{m}''_{\mathrm{ij}}$ must be in interval [0, 1], the impact of dimension is removed.

$$M'' = \begin{pmatrix} m''_{11} & m''_{12} & m''_{13} & \ldots & m''_{1j} \\ m''_{21} & m''_{22} & m''_{23} & & \vdots \\ m''_{31} & m''_{32} & m''_{33} & & \vdots \\ \vdots & & & \ddots & \vdots \\ m''_{i1} & \ldots & \ldots & \ldots & m''_{ij} \end{pmatrix} \tag{3}$$

2.1.3 Creating Fuzzy Similarity Matrix

The form of similarity relation matrix (degree of membership matrix) ***R*** is written as follows:

$$R = \begin{pmatrix} r_{11} & & & & \\ r_{21} & r_{22} & & & \\ r_{31} & r_{32} & r_{33} & & \\ \vdots & & & \ddots & \\ r_{n1} & \cdots & \cdots & \cdots & r_{nn} \end{pmatrix} \tag{4}$$

The relation between $\boldsymbol{x}_{\mathrm{i}}$ and $\boldsymbol{x}_{\mathrm{j}}$ is the same as the relation of $\boldsymbol{x}_{\mathrm{j}}$ and $\boldsymbol{x}_{\mathrm{i}}$ (i, j = 1, 2, 3, …, n), so we just need half of a matrix which is divided by diagonal. In addition, any element $\boldsymbol{x}_{\mathrm{i}}$ is the same as itself. So the form of similar relation matrix ***R*** is expressed as

$$R = \begin{pmatrix} 1 & & & & \\ r_{21} & 1 & & & \\ r_{31} & r_{32} & 1 & & \\ \vdots & & & \ddots & \\ r_{n1} & \cdots & \cdots & r_{n(n-1)} & 1 \end{pmatrix} \tag{5}$$

where if the $\boldsymbol{r}_{\mathrm{ij}}$ is "1", $\boldsymbol{x}_{\mathrm{i}}$ and $\boldsymbol{x}_{\mathrm{j}}$ are exactly the same ones, or else if the $\boldsymbol{r}_{\mathrm{ij}}$ is the "0", $\boldsymbol{x}_{\mathrm{i}}$ and $\boldsymbol{x}_{\mathrm{j}}$ are exactly different. In here, we use max-min method to calculate the $\boldsymbol{r}_{\mathrm{ij}}$ and it is shown in the following:

$$r_{ij} = \frac{\sum_{h=1}^{k} \min(m''_{ih}, m''_{jh})}{\sum_{h=1}^{k} \max(m''_{ih}, m''_{jh})} \tag{6}$$

where i < j, because we just need half of the matrix which is divided by diagonal.

2.2 Fuzzy Clustering Algorithm

In the graphic algorithm which is structured by Clustering Analysis of Maximal Tree method, all of the objects are vertexes. If $\boldsymbol{r}_{\mathrm{ij}} \neq 0$, vertex ***i*** and vertex ***j*** can be connected by a line until the all vertexes are connected. But they cannot produce any circuit, because any two vertexes have one relation. So we need remove the lines of a minimum value which are in produced circuit. At last, we get a Maximal Tree, each side of which has a weight "$\boldsymbol{r}_{\mathrm{ij}}$". By the Threshold-λ, we remove all the side which weight $\boldsymbol{r}_{\mathrm{ij}} < \lambda$. In the remaining vertexes, any connected vertexes are included in the same cluster. So if we want to extract something, we need embed their standardized value in data set. If any vertex can be clustered to a group with the test data according to the threshold λ, it must be a useful data element that we see.

3 The Latent Semantic Indexing for Data Mining

Latent Semantic Indexing (LSI) is an indexing and retrieval method that uses a mathematical technique called Singular Value Decomposition (SVD) to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of texts. LSI is based on the principle that words tend to have similar meanings in the same context. It requires relatively high computational performance and large memory in comparison to other information retrieval techniques [6]. A key feature of LSI is its ability to extract the conceptual content of texts by establishing associations among those terms which occur in similar contexts.

Latent Semantic Indexing can be able to correlate semantically terms in a collection of texts. The method was first applied to texts at Bell Laboratories in the late 1980s, also called Latent Semantic Analysis (LSA), which can uncover the underlying latent semantic structure in the usage of words in a body of texts and extract the meaning of the text in response to user queries, commonly referred to as concept searches. Queries, or concept searches, depend on a set of documents which have undergone LSI will return results that are similar in meaning to the search criteria even if the results do not share specific words with the search criteria [7].

3.1 Singular Value Decomposition

In linear algebra, the singular value decomposition (SVD) is an important factorization of a rectangular real or complex matrix, with many applications in signal processing and statistics. Suppose **A** is an m × n matrix whose entries come from a space, which is either the space of real numbers or the space of complex numbers. Then there exists a factorization of the form

$$A = \mathrm{U}\sum \mathrm{V}^{\mathrm{T}} \tag{7}$$

where **U** is an m × m unitary matrix, the matrix **A** is an m × n diagonal matrix with nonnegative real numbers on the diagonal, and $\mathbf{V}^{\mathbf{T}}$ is an n × n unitary matrix, denotes the conjugate transpose of **V**. Such a factorization is called the singular value decomposition of **A**. The diagonal entries $\mathbf{\Sigma_{ij}}$ of **Σ** are known as the singular values of **A**. It's shown in Fig. 2.

For obtaining simplified vector space, we just need pick up the non-zero singular values of **Σ**. The rank-**r** is the amount of non-zero singular values of **Σ**.

Fig. 2 The singular value decomposition

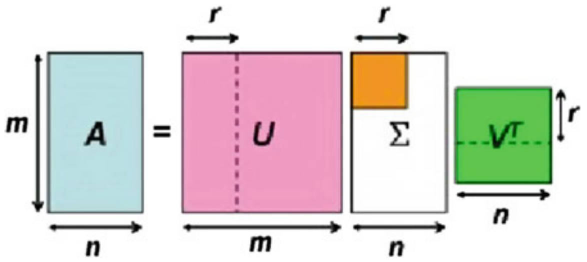
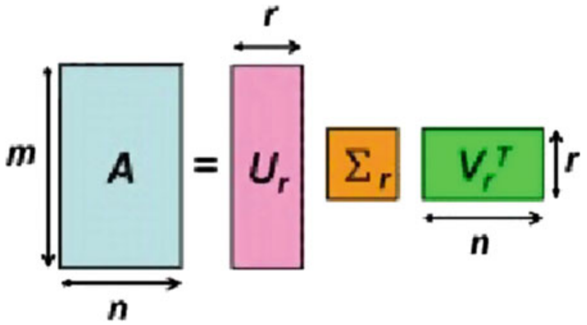


Fig. 3 The simplified SVD



A common usage is to place the singular values in descending order. In this case, the diagonal matrix **Σ** is uniquely determined by **A**.

So we obtain the followed new Eq. (8) and Fig. 3:

$$A = \mathrm{U_r} \sum\nolimits_{\mathrm{r}} V_r^T. \tag{8}$$

3.2 An Example of SVD

This example has terms 1, 2, 3, 4 and 5 as follows:

Term 1 appear 1, 0, 0, 0 times in document 1, 2, 3, 4;
Term 2 appear 0, 0, 0, 4 times in document 1, 2, 3, 4;
Term 3 appear 0, 3, 0, 0 times in document 1, 2, 3, 4;
Term 4 appear 0, 0, 0, 0 times in document 1, 2, 3, 4;
Term 5 appear 2, 0, 0, 0 times in document 1, 2, 3, 4.

Factor matrix A, that is, term-document matrix **A** is shown in matrix (9):

$$A = \begin{array}{c|c} \begin{matrix} t_1 & t_2 & t_3 & t_4 & t_5 \end{matrix} & \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \end{pmatrix} & \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{matrix} \end{array} \tag{9}$$

Step 1:

Calculate the interactive matrix (10).

$$A \cdot A^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 16 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 4 \end{pmatrix} \tag{10}$$

Step 2:

We calculate the eigenvalues and singular vectors of $A \cdot A^T$. The non-zero singular eigenvectors are shown by descending as:

$$\lambda_1 = 16,\ \lambda_2 = 9,\ \lambda_3 = 5.$$

The non-zero singular values are obtained in the following:
$\delta_1 = 4,\ \delta_2 = 3,\ \delta_3 = \sqrt{5}$, the rank-**r** of non-zero singular matrix $\mathbf{\Sigma}$ is 3.
So, the eigenvectors are in the following:

$$v_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix},\ v_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},\ v_3 = \begin{pmatrix} \sqrt{0.2} \\ 0 \\ 0 \\ 0 \\ \sqrt{0.8} \end{pmatrix}.$$

So the singular value matrix $\mathbf{\Sigma}$ and the term decomposition matrix $\mathbf{V}$ are shown as:

$$\Sigma_r = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & \sqrt{5} \end{pmatrix};$$

$$V_r^T = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \sqrt{0.2} & 0 & 0 & 0 & \sqrt{0.8} \end{pmatrix}.$$

Step 3:

Calculating the document decomposition matrix U.

$$u_1 = \frac{1}{\delta_1} A v_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad u_2 = \frac{1}{\delta_2} A v_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad u_3 = \frac{1}{\delta_3} A v_3 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$U_r = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

So if we want to analyze the attribute of document, we just need calculate the document decomposition matrix U_r.

4 Rough Set Based Optimization for Data Mining

Pawlak originally proposed the concept of a rough set as a mathematical approach to handle imprecision, vagueness and uncertainty in data [8]. This method has amply demonstrated its usefulness and versatility through successful applications to a wide variety of problems [9, 10]. The theory behind rough sets involves the approximation of an arbitrary subset of a universe by using two definable or observable subsets called lower and upper approximations. Both approximations in rough set theory may granulate, that is, cluster and express knowledge in the form of decision rules in the context of information systems [11, 12]. In this application, we also use fuzzy clustering to develop a new algorithm for granulating knowledge by means of a seriousness grade.

Table 1 Sample data

No.	Attribute			
	Economic loss A_1	Disaster area A_2	Casualty state A_3	Length of time C_4
1	NG	NG	ES	NG
2	G	NG	I	NG
3	NG	VL	NG	NG

Table 2 The discrimination matrix

	Object 1	Object 2	Object 3
Object 1	–		
Object 2	C_1, C_3	–	
Object 3	C_2, C_3	C_1, C_2, C_3	–

4.1 Discrimination Matrix

The discrimination matrix is created based on the similarities of objects with which the ranking is made. It can present discrimination between any two objects from multiple attribution data. Every element of the discrimination matrix means discrimination of two objects [13]. An example of discrimination matrix is shown as following:

- Objects 1 and 2 can be differentiated by attribute A_1 or A_3;
- Objects 1 and 3 can be discriminated by attribute A_2 or A_3;
- Objects 2 and 3 can be differentiated by means of attribute $A_{1,}$ A_2 or A_3.

As Table 1 shows, the discrimination matrix illustrates the discrimination among objects.

Table 2 enables us to derive the following attribute logical relation:

$$\Delta = (A_1 \vee A_3) \wedge (A_2 \vee A_3) \wedge (A_1 \vee A_2 \vee A_3) = (A_1 \vee A_3) \wedge (A_2 \vee A_3)$$

where $\wedge$ and $\vee$ denote disjunction and conjunction logical operations, respectively. $(A_1 \vee A_3) \wedge (A_2 \vee A_3)$ is Disjunctive Normal Form (DNF) of the attribute logical relation.

4.2 Attribute Reduction Matrix (ARM)

4.2.1 Attribute Reduction Based on the Discrimination Matrix

The discrimination matrix is used to derive the reduction matrix through the following three steps: First, creating the discrimination matrix by definition;

Second, finding out the cores of matrix cells, that are, in rough sets analysis, the essential attributes without that we cannot deduct a matrix. E.g.: In the set $\{\{x_1, x_2, x_5\},\{x_2\}, x_2, x_3, x_4\}\}$, $x_1, x_2, x_3, x_4, x_5 \in A$, x_2 is the core of this set. Third calculating, the reduction uses the discrimination function to get the disjunctive normal form (DNF). In this process, the ARM) will bring out a lot of repeated elements(set $C_{ij} = C_{ik}$) or some elements which have contained relationship (or set $C_{ij} \in C_{ik}$). To remove the repeated elements and thus enhance the efficiency of the reduction algorithm, we can use the absorption law to remove ineffective repeated elements.

4.2.2 The Improved Algorithm Based on the Discrimination Matrix

In these steps, $\boldsymbol{c(j, i)}$ is an element of discrimination matrix, $\boldsymbol{b(k)}$ is an attribute, $\boldsymbol{a(i, j)}$ indicates that object $\boldsymbol{i}$ has attribute $\boldsymbol{b(j)}$, $\boldsymbol{d(n)}$ is in disjunctive form, and $\boldsymbol{s}$ is the number of attributes, **D** indicates all of the data, |**D** |denotes the number of objects, $\boldsymbol{m}$ is the number of disjunctive forms, and ***flag*** indicates whether the process uses the absorption law (flag = 1) or not (flag = 0).

Based on attribute reduction of the discrimination matrix, this algorithm is written as follows:

Step 1:

Calculating the full set of attributes **PosC**(D) and equivalence class set **Ind**(D) where **PosC**(D) indicates universal data set, and **Ind**(D) indicates the classified data according to the Decision Table 2.

Step 2:

Calculating the disjunctive normal form (DNF):

(a1) $m = 1$, $\boldsymbol{d}(1) = \boldsymbol{b}(1)\vee\boldsymbol{b}(2)\vee\ldots\ldots\vee\boldsymbol{b}(s)$;

(a2) $i = 1$;

(a3) If $i < |\mathbf{D}|$, then go to (a4), or else go to (Step 3);

(a4) $j = i + 1$;

(a5) If $j \leq |\mathbf{D}|$, then go to (a6), or else go to (a17);

(a6) Set $\boldsymbol{c}(j, i)$ to null;

(a7) If $(x_i \in \mathbf{PosC}(D) \wedge x_j \notin \mathbf{PosC}(D)) \vee (x_i \notin \mathbf{PosC}(D) \wedge x_j \in \mathbf{PosC}(D)) \vee (x_i, x_j \in \mathbf{PosC}(D) \wedge (x_i, x_j) \notin \mathbf{Ind}(D))$, then go to (a8), or else go to (a16);

(a8) $k = 1$;

(a9) If $k \leq s$, then go to (a10), or else go to (a12);

(a10) If $\boldsymbol{a}(j, k) \neq \boldsymbol{a}(i, k)$, then $\boldsymbol{c}(j, i) = \boldsymbol{c}(j, i) \vee \boldsymbol{b}(k)$;

(a11) $k = k + 1$, go to (a9);

(a12) $n = 1$, $\boldsymbol{flag} = 0$;

(a13) If $n \leq m$, then go to (a14), or else go to (a15);

(a14) If d(n) includes $\boldsymbol{c}(j, i)$, then $\boldsymbol{d}(n) = \boldsymbol{c}(j, i)$, $n = n + 1$, $\boldsymbol{flag} = 1$, and go to (a13), or else if $\boldsymbol{d}(n) \subseteq \boldsymbol{c}(j, i)$, then set $\boldsymbol{c(j, i)}$ to Null, go to (a16), or else $n = n + 1$, go to (a13);

(a15) If $n > m$ and ***flag*** $= 0$, then $m = m + 1$, $\boldsymbol{d}(m) = \boldsymbol{c}(j,\ i)$;
(a16) $j = j + 1$, go to (a5);
(a17) $i = i + 1$, go to (a3);

Step 3:

In step 2, for every ***c*(*j*, *i*)**, we judge whether it is included in each ***d*(*n*)**. If it is include, then it is replaced. In the step 2, the element ***c*(*j*, *i*)** of ***d*(*n*)** has inclusion relation. However if ***c*(*j*, *i*)** is included in $\boldsymbol{d}(\boldsymbol{n_1})$ and $\boldsymbol{d}(\boldsymbol{n_2})$, … and ***d*(*n*)** may have the same grade relationship. So we need to remove the repeated elements of ***d*(*n*)**:

(b1) $i = 1$;
(b2) $j = i + 1$;
(b3) If $\boldsymbol{d}(i) = \boldsymbol{d}(j)$, then go to (b4), or else go to (b7);
(b4) $k = j$;
(b5) $\boldsymbol{d}(k) = \boldsymbol{d}(k + 1)$;
(b6) $k = k + 1$, if k < m, then go to (b5), or else $m = m - 1$, go to (b7);
(b7) $j = j + 1$, if $j \leq m$, then go to (b3), or else $i = i + 1$, if $i < m$, then go to (b2), or else go to step 4;

Step 4:

For $i = 1$ to m, we simplify $\boldsymbol{d}(i)$. In the $\boldsymbol{d}(i)$, if the number of attributes is 1, $\boldsymbol{d}(i)$ is the core of discrimination matrix.

Step 5:

Finally, we obtain the reduction set using the discrimination function.

5 The Applications of Data Clustering for Faithful Customer Analysis

Usually, the managers of gas station judge the preference of faithful customers by these two method: average times method and total times method.

(1) **The total value method**:

It means to judge faithful customers by the total consumption value of Petro China. If anyone fills oil, or buys some food, water, etc. above CNY¥1,000 in our gas station every month, he is a faithful customer. Of course, if the consumption of anyone below CNY¥1,000, he is considered as a usual customer only.

(2) **The total times method**:

It means to judge faithful customers by times of filling oil in Petro China. If anyone fills oil in our gas station more than above 30 times, he becomes a faithful customer. Of course, if anyone fills oil 20 times, he is taken for a normal customer only.

But these two methods have some errors. For example, there are some customers: customer A is a salary man, he drives a car everyday for 15 km and fills oil about every 60 days; customer B is a salary man too, he drives a car everyday for 50 km and fills oil about every 20 days; customer C is a driver of an express company, he drives car everyday for 200 km and fills oil about every 5 days; customer D is a coach driver, he drives a car everyday for 600 km and fills oil almost every days.

The customer A is a faithful customer of our gas station and he fills oil in Petro China gas station only. So, in a year, he fills oil about 6 times in all and once every 60 days on average.
The customer B is a normal customer of our gas station and sometime she fills oil in Petro China gas station. So, in a year, he fills oil about 10 times in all and once every 40 days on average.
The customer C is a normal customer of our gas station and sometime he fills oil in Petro China gas station. So, in a year, he fills oil about 35 times in all and once every 10 days on average.
The customer D is faithful customer of other gas station and he fills oil in Petro China gas station seldom. So, in a year, he fills oil about 30 times in all and once every 12 days on average.

So by these two methods: The customer C and D are faithful customer of Petro China gas station. The customer B is a normal customer of Petro China gas station. The customer A is not a faithful customer of Petro China gas station. In fact, the analysis results are almost all wrong except customer B.

5.1 Rough Set Based LSI for Analysis of Faithful Customer

5.1.1 The Production Process of the Maximal Tree

The performance of this novel method was evaluated using a test database comprised of customers. Table 3 shows the test data.

First, we must to standardize the data, as shown in Table 4. It is not easy to derive the discrimination matrix from these data. Rather, we must transform it into a decision table. Using the fuzzy grade (Table 5), we transformed Table 4 to decision table (Table 6). Based on this decision table, we obtain the equivalence relation **Ind**(D) = $\{\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18\}, \{19, 20, 21, 22, 23, 24\}, \{25, 26, 27, 28\}\}$ and thus evaluate the data set **PosC**(D) = $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28\}$.

From Table 6, we can build the discrimination matrix. Customers 1 and 2 can be discriminated by attributes A_1 and A_3, $r_{1,2} = \{A_1, A_3\}$, and Customers 1 and 3 can be discriminated by attributes A_2 and A_3, $r_{1,3} = \{A_2, A_3\}$. In addition, Customers 3 and 4 can be discriminated by attributes A_1, A_2 and A_3, $r_{1,4} = \{A_1, A_2, A_3\}$, but $r_{1,3}$ $r_{1,4}$, and so $r_{1,4}$ is absorbed. Thus we set it to null. Customers 1 and 7

Table 3 Test data

No.	Attribute				Result
	Gas A_1(CNY¥)	Auto parts A_2(CNY¥)	Water A_3(CNY¥)	Foods A_4(CNY¥)	
1	74	87	236	3	Faithful
2	1,123	25	120	13	Faithful
3	213	1,135	10	1	Faithful
4	1,213	345	46	5	Faithful
5	1,213	1,134	9	19	Faithful
6	634	645	5	3	Faithful
7	115	535	5	9	Faithful
8	288	342	54	13	Faithful
9	67	1,687	1	5	Faithful
10	636	610	6	7	Faithful
11	1,567	55	62	2	Faithful
12	167	967	39	12	Faithful
13	1,334	20	56	8	Faithful
14	178	714	57	17	Faithful
15	768	44	7	6	Faithful
16	2,713	2	6	9	Faithful
17	371	43	185	13	Faithful
18	527	832	55	16	Faithful
19	51	15	52	28	General
20	7	421	0	1	General
21	68	81	50	26	General
22	610	73	12	3	General
23	76	392	15	36	General
24	13	323	7	72	General
25	66	18	3	9	Other
26	3	35	0	12	Other
27	25	39	1	9	Other
28	18	20	5	3	Other

can be discriminated by attributes A_2 and A_3, $r_{1,7} = \{A_2, A_3\}$, but $r_{1,3} = r_{1,7}$, and so the $r_{1,7}$ is absorbed. Thus, we set it to null. We pursued the procedure like this to the last one $r_{27,28} = \varnothing$. We obtain the discrimination matrix (Table 7).

Using the (ARM) algorithm, we obtain $\boldsymbol{d}(1) = A_1 \vee A_3$, $\boldsymbol{d}(2) = A_2 \vee A_3$, $\boldsymbol{d}(3) = A_3$, $\boldsymbol{d}(4) = A_1 \vee A_2$, $\boldsymbol{d}(5) = A_1 \vee A_4$, $\boldsymbol{d}(6) = A_2$, and $\boldsymbol{d}(7) = A_1$. The core is A_1, A_2, and A_3. By the absorption law, we obtained the following discrimination functions as:

$$\begin{aligned} \varDelta &= A_1 \wedge A_2 \wedge A_3 \wedge (A_1 \vee A_3) \wedge (A_2 \vee A_3) \wedge (A_1 \vee A_2) \wedge (A_1 \vee A_4) \\ &= A_1 \wedge A_2 \wedge A_3 \end{aligned}$$

Table 4 Standardized data

No.	Attribute				Result
	Gas	Auto parts	Water	Foods	
	A_1	A_2	A_3	A_4	
1	0.0262	0.0504	1.0000	0.0417	Faithful
2	0.4133	0.0136	0.5085	0.1806	Faithful
3	0.0775	0.6724	0.0424	0.0139	Faithful
4	0.4465	0.2036	0.1949	0.0694	Faithful
5	0.4465	0.6718	0.0381	0.2639	Faithful
6	0.2328	0.3816	0.0212	0.0417	Faithful
7	0.0413	0.3163	0.0212	0.1250	Faithful
8	0.1052	0.2018	0.2288	0.1806	Faithful
9	0.0236	1.0000	0.0042	0.0694	Faithful
10	0.2336	0.3608	0.0254	0.0972	Faithful
11	0.5771	0.0315	0.2627	0.0278	Faithful
12	0.0605	0.5727	0.1653	0.1667	Faithful
13	0.4911	0.0107	0.2373	0.1111	Faithful
14	0.0646	0.4226	0.2415	0.2361	Faithful
15	0.2823	0.0249	0.0297	0.0833	Faithful
16	1.0000	0.0000	0.0254	0.1250	Faithful
17	0.1358	0.0243	0.7839	0.1806	Faithful
18	0.1934	0.4926	0.2331	0.2222	Faithful
19	0.0177	0.0077	0.2203	0.3889	General
20	0.0015	0.2487	0.0000	0.0139	General
21	0.0240	0.0469	0.2119	0.3611	General
22	0.2240	0.0421	0.0508	0.0417	General
23	0.0269	0.2315	0.0636	0.5000	General
24	0.0037	0.1905	0.0297	1.0000	General
25	0.0232	0.0095	0.0127	0.1250	Other
26	0.0000	0.0196	0.0000	0.1667	Other
27	0.0081	0.0220	0.0042	0.1250	Other
28	0.0055	0.0107	0.0212	0.0417	Other

We obtain reduction-standardized data, as shown in Table 8. The classification of the test data and the mining of the test data are necessary to judge the quantity of gas(A_1), quantity of auto parts(A_2) and quantity of water(A_3) only of accidents. In Table 8, we removed the length of time in the context of accidents, which had been included in Table 6. The number of elements in the process has been cut about 25 % off. We need judge only 3 attributes of all 4 attributes.

5.1.2 Improved Fuzzy Clustering

We can now use the data from attribute reduction for improved fuzzy mining. We aim to extract two special types of customers for future research. But how to extract them is the purpose of this experiment. In the first type, we want to extract

Table 5 The fuzzy grade dictionary

Attribute		Linguistic variables	Fuzzy grade
Gas	A_1	Nothing(NG)	[0.00, 0.25)
		Mild(M)	[0.20, 0.40)
		Great(G)	[0.40, 0.60)
		Very great(VG)	[0.60, 0.80)
		Extremely great(EG)	(0.80, 1.00]
Auto parts	A_2	Nothing(NG)	[0.00, 0.25)
		Mild(M)	[0.20, 0.40)
		Great(G)	[0.40, 0.60)
		Very great(VG)	[0.60, 0.80)
		Extremely great(EG)	(0.80, 1.00]
Water	A_3	Nothing(NG)	[0.00, 0.25)
		Mild(M)	[0.20, 0.40)
		Great(G)	[0.40, 0.60)
		Very great(VG)	[0.60, 0.80)
		Extremely great(EG)	(0.80, 1.00]
Foods	A_4	Nothing(NG)	[0.00, 0.25)
		Mild(M)	[0.20, 0.40)
		Great(G)	[0.40, 0.60)
		Very great(VG)	[0.60, 0.80)
		Extremely great(EG)	(0.80, 1.00]

some customers who don't buy any water, but fill gas and buy some auto parts. The grade of gas is 0.5, the grade of auto parts is 0.6, and the grade of water is 0. In the second type, we want to extract some customers which like grade of gas is 0.5, the grade of auto parts is 0, and the grade of water is 0.5. We embed these two new sets of test data (the 19th and 20th objects in Table 9), and the new data set is shown in Table 9.

In Table 9, the evaluated objects are labeled No. 19 and No. 20. We use the improved fuzzy clustering method to extract all relevant data elements. Details are in Table 9. We remove all the data of "General" grade and "Other" grade, because these data cannot be clustered with "Faithful" grade.

Step 1:

We use the max-min method to calculate the weight $\boldsymbol{r}_{\mathrm{ij}}$, as shown in Table 10. Any element $\boldsymbol{x}_{\mathrm{i}}$ is the same as itself, and so all the diagonal elements are "1".

Step 2:

We use similarity data to build the maximal tree. First, we pick all the vertices for which relation values are not less than 0.6. Thus, the vertices {1, 17}, {2, 11}, {2, 13}, {2, 20}, {3, 5}, {3, 9}, {3, 12}, {4, 11}, {4, 13}, {5, 19}, {6, 7}, {6, 10}, {6, 18}, {7, 10}, {8, 14}, {10, 18}, {11, 13}, {11, 20}, {12, 14}, {12, 18}, {13, 20} and {14, 18} are selected up. We connect these vertices by their similarity value to build the initial cluster tree, as shown in Fig. 4.

Table 6 Decision table

No.	Attribute				Result
	Gas C_1	Auto parts C_2	Water C_3	Foods C_4	
1	NG	NG	EG	NG	Faithful
2	G	NG	G	NG	Faithful
3	NG	VG	NG	NG	Faithful
4	G	M	NG	NG	Faithful
5	G	VG	NG	M	Faithful
6	M	M	NG	NG	Faithful
7	NG	M	NG	NG	Faithful
8	NG	M	M	NG	Faithful
9	NG	EG	NG	NG	Faithful
10	M	M	NG	NG	Faithful
11	G	NG	M	NG	Faithful
12	NG	G	NG	NG	Faithful
13	G	NG	M	NG	Faithful
14	NG	G	M	M	Faithful
15	M	NG	NG	NG	Faithful
16	EG	NG	NG	NG	Faithful
17	NG	NG	VG	NG	Faithful
18	NG	G	M	M	Faithful
19	NG	NG	M	M	General
20	NG	M	NG	NG	General
21	NG	NG	M	M	General
22	M	NG	NG	NG	General
23	NG	M	NG	G	General
24	NG	NG	NG	EG	General
25	NG	NG	NG	NG	Other
26	NG	NG	NG	NG	Other
27	NG	NG	NG	NG	Other
28	NG	NG	NG	NG	Other

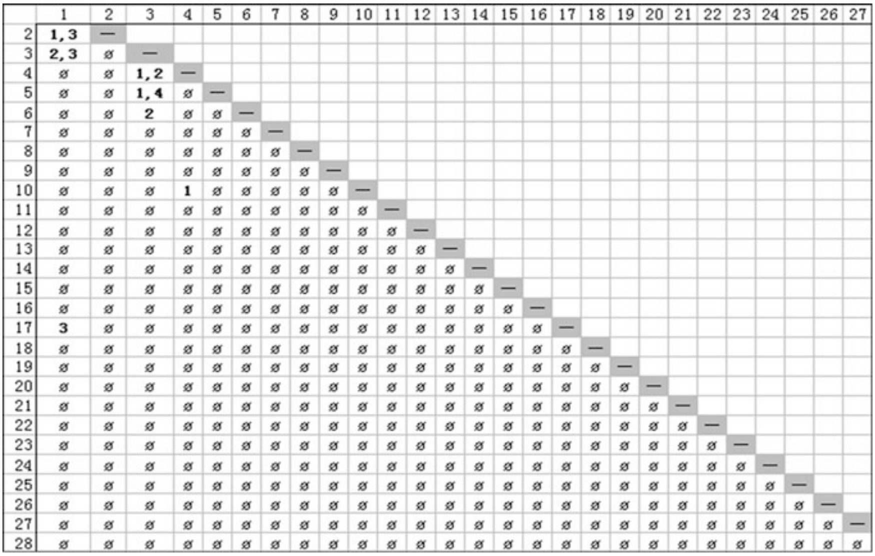
Step 3:

However vertices 15 and 16 are not included in the cluster tree. We thus need to add these two vertices and connect vertices with maximum similarity value of them. The similarity values are 0.43 for {13, 15}, 0.47 for {11, 16} and 0.57 for {4, 8}. So, all the vertices are connected.

Step 4:

The maximal tree cannot have a circuit because any two vertices have only one maximal relation. Thus, we must remove the edge of minimum value in a circuit. The edge {4, 11}, {2, 11}, {2, 13}, {12, 18}, {6, 7}, {10, 18}, and {11, 20} should be removed. The final maximal tree is shown in Fig. 5.

Table 7 Discrimination matrix



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
2	1,3	—																									
3	2,3	ø	—																								
4	ø	ø	1,2	—																							
5	ø	ø	1,4	ø	—																						
6	ø	ø	2	ø	ø	—																					
7	ø	ø	ø	ø	ø	ø	—																				
8	ø	ø	ø	ø	ø	ø	ø	—																			
9	ø	ø	ø	ø	ø	ø	ø	ø	—																		
10	ø	ø	ø	1	ø	ø	ø	ø	ø	—																	
11	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	—																
12	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	—															
13	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	—														
14	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	—													
15	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	—												
16	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	—											
17	3	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	—										
18	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	—									
19	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	—								
20	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	—							
21	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	—						
22	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	—					
23	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	—				
24	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	—			
25	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	—		
26	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	—	
27	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	—
28	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø

5.1.3 Data Mining of Fuzzy Clustering

If we wish to extract accident cases of high similarity grade, we can select a similarity value of $\lambda = 0.8$. After removing all edges for which similarity values are below 0.8, Customer 5 is the same as Customer 19. Customer 2 is the same as Customer 20. In addition, Customer 5 is similar to Customer 19, and Customer 2 resembles Customer 20. Customer 2 and 5 are the Customer s that we wish to extract. The result is shown in Fig. 6.

If we select a similarity value of $\lambda = 0.65$, based on the clustering tree, Customers 3, 5, 12, 14 and 18 are similar to Customer 19, whereas Customers 2, 4, 11 and 13 are similar to Customer 20. Thus, Customers 3, 5, 12, 2, 4, and 11 comprise the total required Customers for our purpose. The result is shown in Fig. 7.

Of course, we can extract them by individual approximation. For example, we can set extracted condition which like $\{L_{C1} \leq C_1 \leq H_{C1}$ and $L_{C2} \leq C_2 \leq H_{C2}$ and $L_{C3} \leq C_3 \leq H_{C3}\}$ and $\{L_{C1} \leq C_1 \leq H_{C1}$ or $L_{C2} \leq C_2 \leq H_{C2}$ or $L_{C3} \leq C_3 \leq H_{C3}\}$. The L_{Cn} means low limit of attribute C_n. The H_{Cn} means high limit of attribute C_n. But if we use "and" relationship, we must ignore some cases which have a lot of similar attributes and only one or two dissimilar attributes. If we use "or" relationship, we must extract some noise cases which have only one or two similar attributes and a lot of dissimilar attributes.

Table 8 Reduction standardized data

No.	Attribute			Result
	Gas A_1	Auto parts A_2	Water A_3	
1	0.0262	0.0504	1.0000	Faithful
2	0.4133	0.0136	0.5085	Faithful
3	0.0775	0.6724	0.0424	Faithful
4	0.4465	0.2036	0.1949	Faithful
5	0.4465	0.6718	0.0381	Faithful
6	0.2328	0.3816	0.0212	Faithful
7	0.0413	0.3163	0.0212	Faithful
8	0.1052	0.2018	0.2288	Faithful
9	0.0236	1.0000	0.0042	Faithful
10	0.2336	0.3608	0.0254	Faithful
11	0.5771	0.0315	0.2627	Faithful
12	0.0605	0.5727	0.1653	Faithful
13	0.4911	0.0107	0.2373	Faithful
14	0.0646	0.4226	0.2415	Faithful
15	0.2823	0.0249	0.0297	Faithful
16	1.0000	0.0000	0.0254	Faithful
17	0.1358	0.0243	0.7839	Faithful
18	0.1934	0.4926	0.2331	Faithful
19	0.0177	0.0077	0.2203	General
20	0.0015	0.2487	0.0000	General
21	0.0240	0.0469	0.2119	General
22	0.2240	0.0421	0.0508	General
23	0.0269	0.2315	0.0636	General
24	0.0037	0.1905	0.0297	General
25	0.0232	0.0095	0.0127	Other
26	0.0000	0.0196	0.0000	Other
27	0.0081	0.0220	0.0042	Other
28	0.0055	0.0107	0.0212	Other

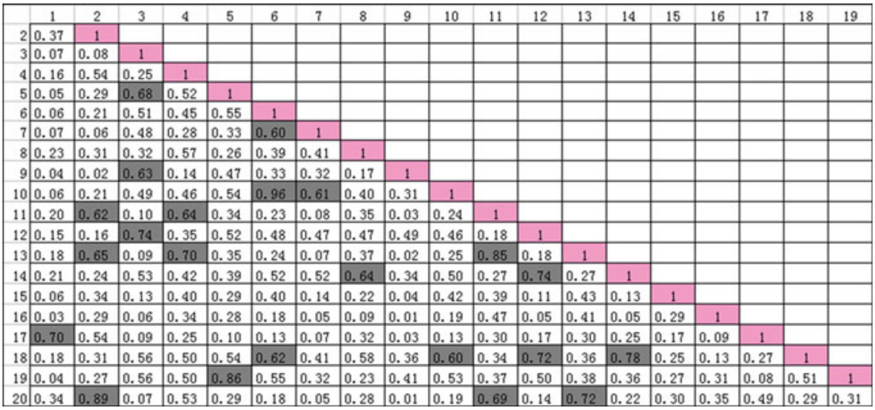
5.2 Customer Clustering Based on LSI

Latent semantic analysis employs information retrieval to explore latent semantics and consumption of customers. Therefore, we use the latent semantic method to analyze big data of gas station customers. This analysis has the following steps: using rough set for optimization, build the primary matrix, execute singular value decomposition, purse similarity analysis of customers' information and obtain the results of clustering [14].

Table 9 Final standardization data

No.	Attribute			Result
	Gas A_1	Auto parts A_2	Water A_3	
1	0.0262	0.0504	1.0000	Faithful
2	0.4133	0.0136	0.5085	Faithful
3	0.0775	0.6724	0.0424	Faithful
4	0.4465	0.2036	0.1949	Faithful
5	0.4465	0.6718	0.0381	Faithful
6	0.2328	0.3816	0.0212	Faithful
7	0.0413	0.3163	0.0212	Faithful
8	0.1052	0.2018	0.2288	Faithful
9	0.0236	1.0000	0.0042	Faithful
10	0.2336	0.3608	0.0254	Faithful
11	0.5771	0.0315	0.2627	Faithful
12	0.0605	0.5727	0.1653	Faithful
13	0.4911	0.0107	0.2373	Faithful
14	0.0646	0.4226	0.2415	Faithful
15	0.2823	0.0249	0.0297	Faithful
16	1.0000	0.0000	0.0254	Faithful
17	0.1358	0.0243	0.7839	Faithful
18	0.1934	0.4926	0.2331	Faithful
19	0.5000	0.6000	0.0000	–
20	0.5000	0.0000	0.5000	–

Table 10 Similarity data



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
2	0.37	1																	
3	0.07	0.08	1																
4	0.16	0.54	0.25	1															
5	0.05	0.29	0.68	0.52	1														
6	0.06	0.21	0.51	0.45	0.55	1													
7	0.07	0.06	0.48	0.28	0.33	0.60	1												
8	0.23	0.31	0.32	0.57	0.26	0.39	0.41	1											
9	0.04	0.02	0.63	0.14	0.47	0.33	0.32	0.17	1										
10	0.06	0.21	0.49	0.46	0.54	0.96	0.61	0.40	0.31	1									
11	0.20	0.62	0.10	0.64	0.34	0.23	0.08	0.35	0.03	0.24	1								
12	0.15	0.16	0.74	0.35	0.52	0.48	0.47	0.47	0.49	0.46	0.18	1							
13	0.18	0.65	0.09	0.70	0.35	0.24	0.07	0.37	0.02	0.25	0.85	0.18	1						
14	0.21	0.24	0.53	0.42	0.39	0.52	0.52	0.64	0.34	0.50	0.27	0.74	0.27	1					
15	0.06	0.34	0.13	0.40	0.29	0.40	0.14	0.22	0.04	0.42	0.39	0.11	0.43	0.13	1				
16	0.03	0.29	0.06	0.34	0.28	0.18	0.05	0.09	0.01	0.19	0.47	0.05	0.41	0.05	0.29	1			
17	0.70	0.54	0.09	0.25	0.10	0.13	0.07	0.32	0.03	0.13	0.30	0.17	0.30	0.25	0.17	0.09	1		
18	0.18	0.31	0.56	0.50	0.54	0.62	0.41	0.58	0.36	0.60	0.34	0.72	0.36	0.78	0.25	0.13	0.27	1	
19	0.04	0.27	0.56	0.50	0.86	0.55	0.32	0.23	0.41	0.53	0.37	0.50	0.38	0.36	0.27	0.31	0.08	0.51	1
20	0.34	0.89	0.07	0.53	0.29	0.18	0.05	0.28	0.01	0.19	0.69	0.14	0.72	0.22	0.30	0.35	0.49	0.29	0.31

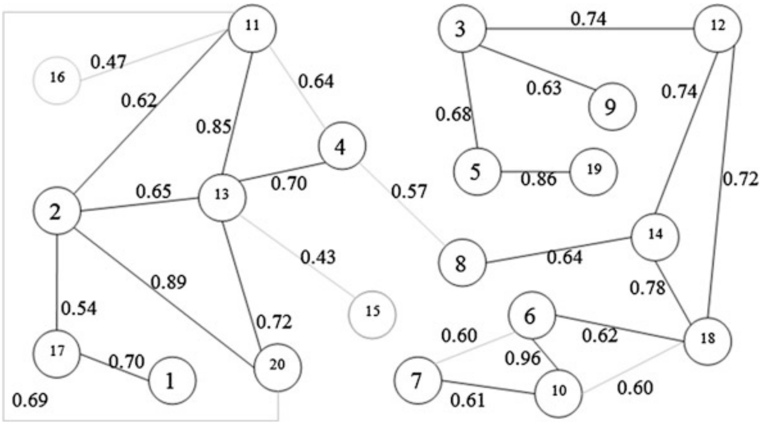


Fig. 4 Initial cluster tree

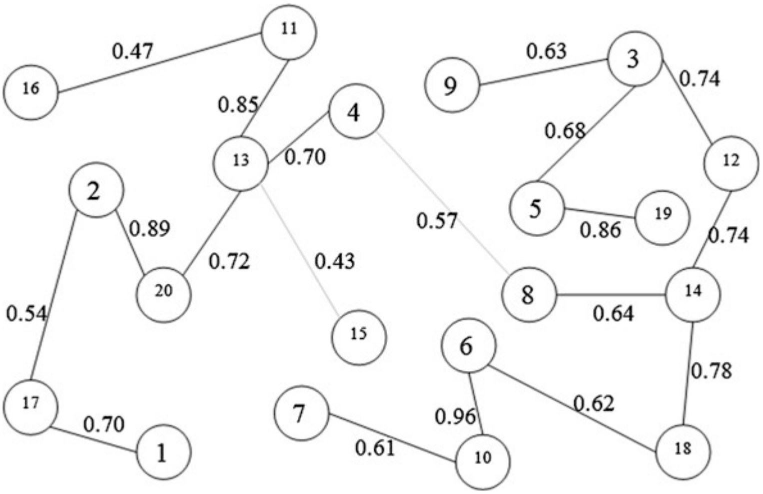


Fig. 5 Final maximal tree

Fig. 6 $\lambda = 0.8$

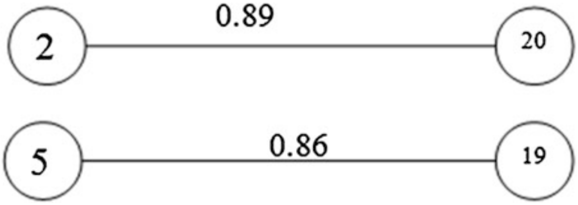


Fig. 7 λ = 0.65

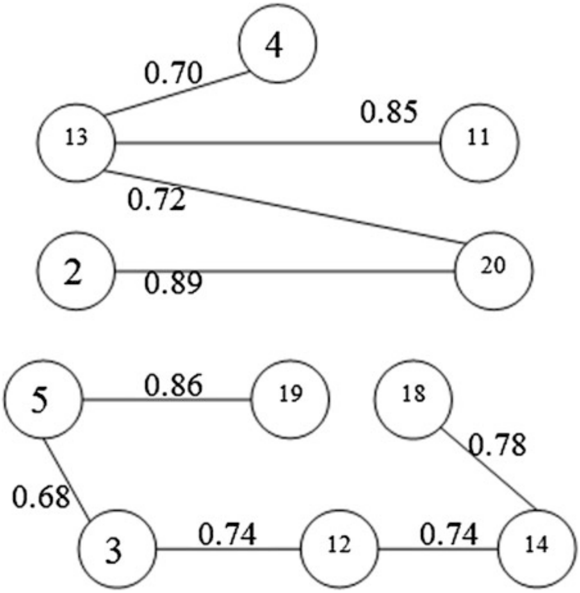
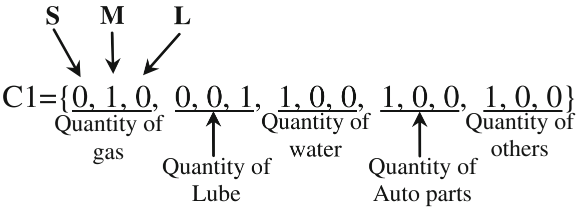


Fig. 8 The description of accident attributes



5.2.1 Establishing Primary Matrix

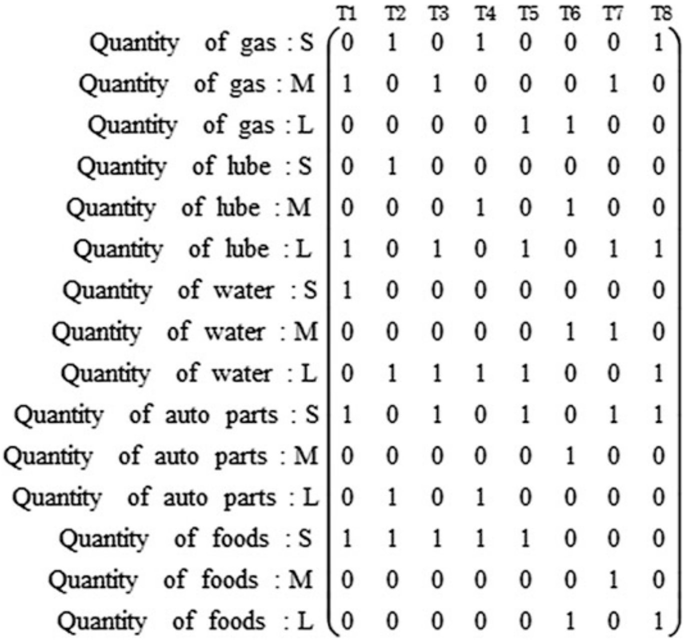
After optimizing of rough set, we ignore the irrespective preferences(cleaning products), and obtain 5 efficient preferences: (1) quantity of gas, (2) quantity of lube, (3) quantity of water, (4) quantity of auto parts and (5) quantity of foods. Every preference has 3 quantity grades which are **S**(small), **M**(medium) and **L**(large). For example, in time 1, a customer fills a medium quantity of gas, buys a large quantity of lube, buys a small quantity of water, buys a small quantity of auto parts, and buys a small quantity of cleaning goods. If any customer attribute has any grade, then this grade is “1”, or else it is “0” and may denote the expression shown in Fig. 8.

Table 11 recorded 8 times consumption (T1–T8) of a customer from the gas station. Thus, we can transform Table 11 to a primary matrix, as shown in Table 12.

Table 11 Sample data

No.	Attribute				
	Gas	Lube	Water	Auto parts	Foods
T1	M	L	S	S	S
T2	S	S	L	L	S
T3	M	L	L	S	S
T4	S	M	L	L	S
T5	L	L	L	S	S
T6	L	M	M	M	L
T7	M	L	M	S	M
T8	S	L	L	S	L

Table 12 The primary matrix



$$\begin{array}{r|cccccccc|}
 & T1 & T2 & T3 & T4 & T5 & T6 & T7 & T8 \\
\text{Quantity of gas : S} & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
\text{Quantity of gas : M} & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
\text{Quantity of gas : L} & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
\text{Quantity of lube : S} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\text{Quantity of lube : M} & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
\text{Quantity of lube : L} & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\
\text{Quantity of water : S} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\text{Quantity of water : M} & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
\text{Quantity of water : L} & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\
\text{Quantity of auto parts : S} & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\
\text{Quantity of auto parts : M} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\text{Quantity of auto parts : L} & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
\text{Quantity of foods : S} & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
\text{Quantity of foods : M} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
\text{Quantity of foods : L} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1
\end{array}$$

5.2.2 Singular Value Decomposition

Table 6.12 shows eight cases with five attributes. The first and second cases are sample data from which we want to extract data, such as this sample below. It creates a 15 × 8 matrix; therefore, we use Matlab to decompose the singular value and obtain the decomposed matrix **U**, **Σ** and $\mathbf{V}^{\mathbf{T}}$. We chose the non-zero singular values of **Σ** to build $\mathbf{U_r}$, $\mathbf{\Sigma_r}$ and $\mathbf{V}_{\mathrm{r}}^{\mathrm{T}}$, which are shown in matrices 5, 6 and 7, respectively.

$$V_r = \begin{pmatrix} -0.22655 & -0.41799 \\ -0.26777 & 0.33835 \\ -0.12346 & 0.000975 \\ -0.\,068385 & -0.2004 \\ -0.093093 & -0.21445 \\ -0.4563 & 0.33619 \\ -0.089526 & 0.11419 \\ -0.094028 & 0.15609 \\ -0.43422 & -0.34894 \\ -0.4563 & 0.33619 \\ -0.02341 & -0.012275 \\ -0.13807 & -0.40257 \\ -0.43527 & -0.21933 \\ -0.070618 & 0.16837 \\ -0.11189 & -0.027689 \end{pmatrix} \tag{11}$$

$$\Sigma_r = \begin{pmatrix} 4.3642 & 0 \\ 0 & 2.8161 \end{pmatrix} \tag{12}$$

$$U_r = \begin{pmatrix} -0.39071 & 0.32158 \\ -0.29845 & -0.56435 \\ -0.46969 & 0.15712 \\ -0.30411 & -0.56933 \\ -0.43633 & 0.037314 \\ -0.10217 & -0.034567 \\ -0.30819 & 0.47413 \\ -0.38615 & -0.043407 \end{pmatrix} \tag{13}$$

5.2.3 The Clustering of Customers

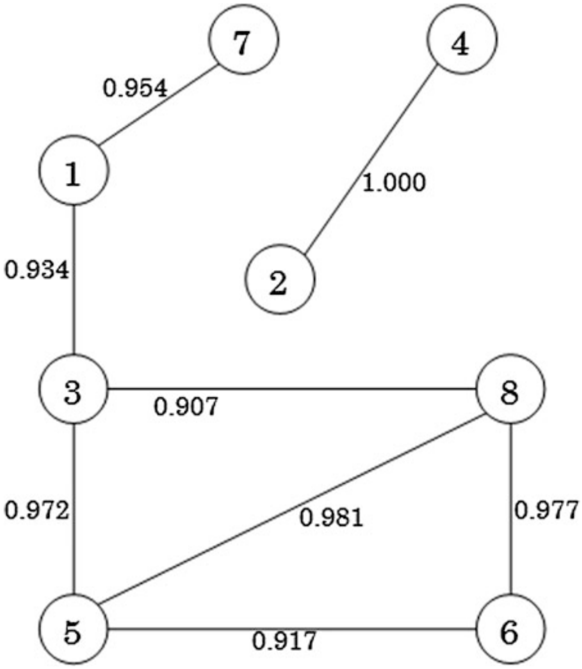
At first, we calculate the cosine of any two customer cases in 2-Dimensional semantic space. If the cosine of these two cases is "1", the angle of two case vectors is 0-degree, that is, the two cases are the same, or else the cosine is "0" and these two cases are not the same. All of the values are shown in Table 13.

From the data shown in Table 13, we use the similarity measure data to build the maximal tree. First, we pick all of the vertices for which relation values are not less than 0.9. Thus, the vertices {1, 3}, {1, 7}, {2, 4}, {3, 5}, {3, 8}, {5, 6}, {5, 8}, and {6, 8} are the selected outputs. We connect these vertices by their similarity value to build the initial cluster tree, as shown in Fig. 9.

Table 13 Fuzzy similarity data

	1	2	3	4	5	6	7	8
1	1							
2	−0.20082	1						
3	0. 93383	0.16291	1					
4	−0.19676	0.99999	0.16699	1				
5	0.82342	0.39052	0.97192	0.39433	1			
6	0.52771	0.72615	0.79565	0.72899	0.91652	1		
7	0.95362	−0.4864	0.78282	−0.48278	0.61441	0.24753	1	
8	0.69629	0.56331	0.90698	0.56673	0.98062	0.97712	0.44792	1

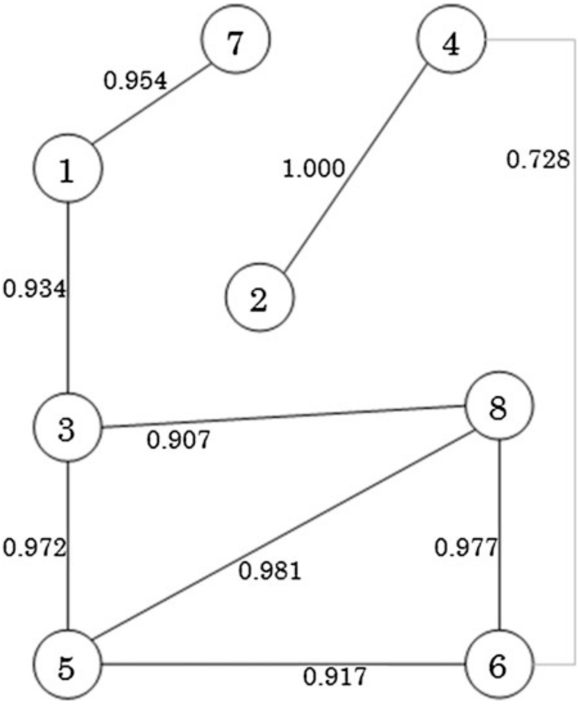
Fig. 9 The initial cluster tree



The maximal tree cannot have a circuit because any two vertices have only one maximal relation. Thus, we must remove the edge of minimum value in a circuit. The edge {3, 8}, and {5, 6} should be removed. The next cluster tree is shown in Fig. 10.

However vertices 2 and 4 are not included in the group{1, 3, 5, 6, 7, 8}. Thus, we must add the two groups of vertices, {2, 4} and {1, 3, 5, 6, 7, 8} and connect the vertices with maximum similarity values between the two groups. The similarity values are 0.782 for {4, 6}. Therefore, all the vertices are connected. The cluster tree is shown in Fig. 11.

Fig. 10 The clustering tree



At last, we obtain a maximal tree, in which each side has a weight "$\boldsymbol{r}_{\mathrm{ij}}$". Using the Threshold-λ, we remove all of the sides that have a weight of $\boldsymbol{r}_{\mathrm{ij}} < \lambda$. In the remaining vertices, any connected vertices are the same cluster [14].

If we select a high similarity value of $\lambda = 0.900$, based on the clustering tree, we delete all of the sides with a similarity value of λ below 0.900. Therefore, the consumption times 1, 3, 5, 6, 7 and 8 are similar which is in cluster 1, and the consumption times 2, 4 are similar which is in cluster 2, as shown in Fig. 12. The cluster 1 has most cluster amounts in all clusters. So it is main preference of this customer. Of course, we can set threshold λ to others for improving the accuracy rate of customer's preference.

Usually, we can obtain preference of customers by judging the total value method of consumption and the total times method of consumptions. But if the products have high price difference, the total value method and total times method are very difficult to judge the preference of customers. For example, if we set anyone spends CNY¥500 for anything, he would like buy it. But CNY¥500 can fill one time gasoline only. If we set anyone buy anything 10 times, he would like buy it. But some customers buy cheap products by large amount and it is very easy to up to 10 times. Of course, we can combine these two methods for judging preference of customers. But it is very difficult to judge multi-preference of customers in a large database by high accuracy rate.

Fig. 11 The maximal tree

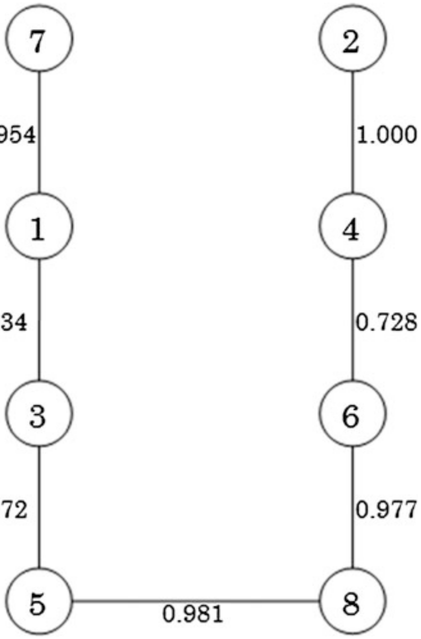


Fig. 12 Clustering by λ = 0.900

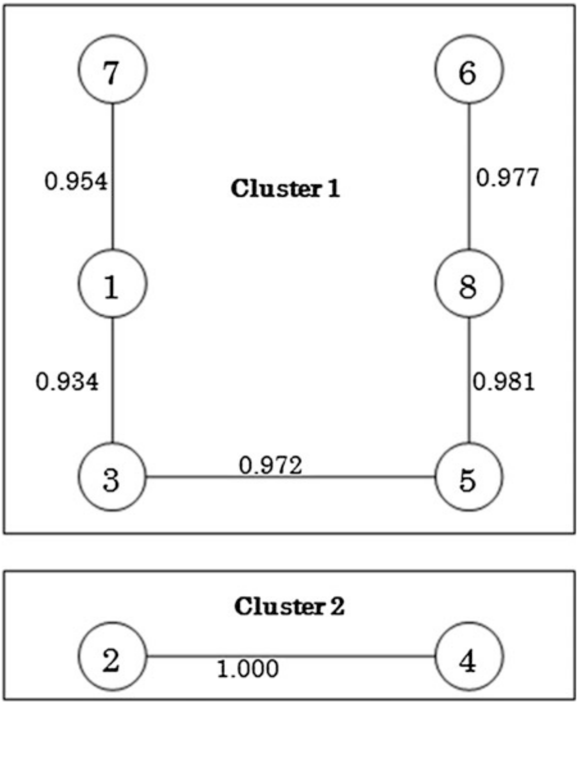
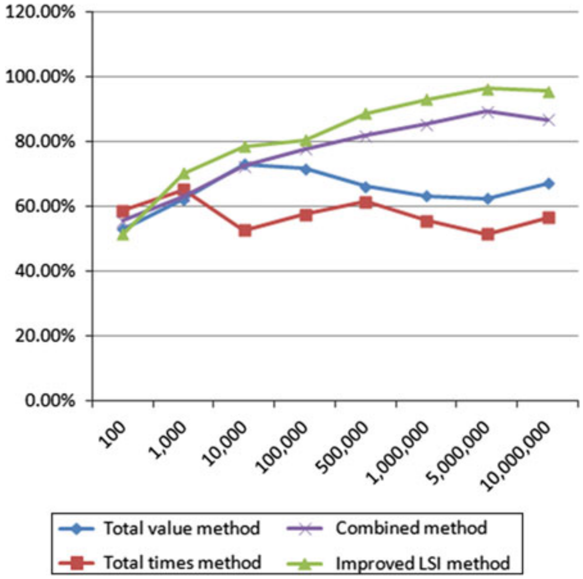


Table 14 A comparison of general methodology and LSI methodology

Data amount	Total value method Accuracy rate	Total times method Accuracy rate	Combined method Accuracy rate	Improved LSI method Accuracy rate
100	53.18 %	58.78 %	55.58 %	51.61 %
1,000	62.32 %	65.25 %	63.35 %	70.37 %
10,000	73.15 %	52.73 %	72.51 %	78.61 %
100,000	71.68 %	57.55 %	77.83 %	80.55 %
500,000	66.27 %	61.51 %	81.81 %	88.71 %
1,000,000	63.36 %	55.73 %	85.39 %	93.13 %
5,000,000	62.58 %	51.58 %	89.35 %	96.31 %
10,000,000	67.18 %	56.65 %	86.86 %	95.65 %

Fig. 13 The comparison of general methodology and LSI methodology



Different customers have different preference of consumptions, and different scenarios can find different customers. Therefore, we would better use LSI methodology to cluster similar data automatically.

Table 14 and Fig. 13 show that the accuracy rate of Improved LSI method is approximately the same as Total value method, Total times method and Combined method below 100,000 data points. All of them have about 50–70 % accuracy rate. But above that mark, the accuracy rate of the LSI method increases compared to the general methods appreciably. The 100,000 data points is 80.55 %, the 500,000 data points is 88.71 %, the 1,000,000 data points is 93.13 %, the 5,000,000 data points is 96.31 % and the 10,000,000 data points is 95.65 %. But the accuracy rates of general methods are still around 65 % only. We apply rough set to customer resource management (CRM) through the attribute reduction. Rough set do not loss any original classification information, but it reduce irrespective preferences(cleaning products) which is in 6 attributes. So we can raise data analysis speed by 16.67 %. Especially, in the test of 8 months, our gas stations judge preference of faithful customers by the LSI methodology and provided accurate services to these faithful customers to obtain their loyalty. The sales of these gas stations grew approximately 30 % more than the same period of last year on average.

6 Concluding Remarks

Based on the attribute reduction matrix (ARM), after building the discrimination matrix, we apply the absorption law to simplify the discrimination function. After the matrix is built, we can reduce the discrimination matrix using the absorption law. The results show that we can remove some outlying data elements. As such, we need only cluster the reduced data elements.

Fuzzy clustering is the process of dividing data elements into clusters so that items in the same clusters are as similar as possible while items across different clusters are as dissimilar as possible. We can use this clustering method to cluster the data elements which we wish.

The method is useful as a graphic algorithm in obtaining fuzzy rules from data. Fuzzy clustering can help users in the difficult task of data mining and data classification. This threshold is used in the algorithm to select appropriate clusters for the data under consideration. The improved mining method is beneficial in deriving both clusters and the attribute reduction matrix.

The latent semantic indexing (LSI) methodology for information retrieval has applied the singular value decomposition to identify an eigenfunction for a large matrix, whose cells represent the occurrence of terms (or conditions) within documents. This methodology was used to rank text documents and data clustering, based on their relevance to a topic. In this paper, we proposed the method that integrated the Latent Semantic Indexing (LSI) concept to our document clustering. This involves the use of Singular Value Decomposition (SVD) which creates a new abstract and uses a way of finding simplified document collection in matrix representation, so that it could identify the terms and documents which are similar. The considered synthetic and real-world examples demonstrated the improved mining properties due to the befitting cluster and the algorithms capability of determining a suitable similarity degree of clusters in the data.

References

1. Lei, Y., Uren, V., Motta, E.: SemSearch: a search engine for the semantic web. In: Proceedings of 15th Int'l Conference Managing Knowledge in a World of Networks (EKAW '06), pp. 238–245 (2006)
2. Pang-Ning, T., Steinbach, M., Kumar, V.: Introduction to data mining. pearson international ed. 2006 Pearson Education, Inc, New Jersey (2006)
3. Hearst, M.A., Pedersen, J.O.: Reexamining the cluster hypothesis 1996. In: Proceeding of SIGIR '96 (1996)
4. Yang, J.X., Watada, J.: Efficient data mining method based on fuzzy clustering. In: International Symposium on Management Engineering 2010, pp. 242–249, Japan, Kitakyusyu 26–28 Aug 2010
5. Yang, J.X., Watada, J.: Fuzzy clustering Analysis of data mining: application to an accident mining system. Int. J. Innovative Comput. Inf. Control **8**(8), 5715–5724 (2012)
6. Karypis, G., Han, E.: Fast supervised dimensionality reduction algorithm with applications to document categorization and retrieval. In: Proceedings of CIKM-00, 9th ACM Conference on Information and Knowledge Management, pp. 12–19 (2000)
7. Yang, J.X., Watada, J.: Wise search engine based on LSI", L. Cao et al. (eds.) Agents and Data Mining Interaction Lecture Notes in Computer Science, 2010, LNCS Vol.5980/ 2010, pp. 126–136, doi:10.1007/978-3-642-15420-1_11
8. Pawlak, Z., Grzymala-Busse, J., Slowinski, R., Ziarko, W.: Rough sets. Commun. ACM **38**(11), 248–254 (1995)
9. Skowron, A., Polkowski, L.: Rough sets in knowledge discovery. Stud. Fuzziness Soft Comput. **19**, 130–135(1998)
10. Ziarko, W.P, Van Rijsbergen, C.J.: Rough sets, fuzzy sets and knowledge discovery. In: Proceedings, Workshop in Computing, pp. 476–482, London, UK (1994)
11. Imai, S., Watada, J.: Rough sets approach to human resource development of it corporations. In: Witold, P., Shyi-Ming, C. (ed.) New Volume on Granular Computing, Springer, Berlin (2011)
12. Kryszkiewicz, M.: Rough set approach to incomplete information systems. Inf. Sci. **112**(1–4), 39–49 (1998)
13. Yang, J.X., Watada, J.: Rough set based optimization for data mining: an improved fuzzy clustering approach. SICE J. Control Meas. Syst. Integr. **5**(4), 001–008 (2012)
14. Yang, J.X., Watada, J.: Wise search engine based on LSI, 2010 autonomous agents and multiagent systems (AAMAS2010), Workshops-ADMI, Canada, Toronto, pp. 127–137 (2010)

The Property of Different Granule and Granular Methods Based on Quotient Space

Yan-ping Zhang, Ling Zhang and Chenchu Xu

Abstract Nowadays, we have entered the era of big data, and we have to deal with complex systems and massive data frequently. Facing complicated objects, how to describe or present objects is the base to solve questions frequently. So we suppose that a problem solving space, or a problem space for short, is described by a triplet (X, f, Γ), and assume that X is a domain, R is an equivalence relation on X, Γ is a topology of X, [X] is a quotient set under R. Regarding [X] as a new domain, we have a new world to analyse and to research this object, consequently we describe or present a question into different granule worlds, these granular worlds are called the quotient space. Further we are able to predigest and solve a question, i.e. we apply quotient space and granulate to represent an object. Comparing rough set and decision-making tree, the quotient space has the stronger representation. Not only it can represent vectors of the problem domain, different structures between vectors, but also it can define different attribute functions and operations etc. In this paper, we discuss the method how to represent and to partition an object in granular worlds, and educe the relationship of different granular worlds and confirm the degree of granule. We will prove three important theorems of different granules, i.e. to preserve false property theorem and to preserve true property theorem. To solve a problem in different granular worlds, the process procedure of quotient approximate will be applied. We also supply an example of solving problem by different granule worlds—the shortest path of a complex network. The example indicates that to describe or present a complicated object is equal to construct quotient space. In quotient set [X], the complexity to solve a problem is lower than X. We have a new solution method to analysis a big data based on the quotient space theory.

Y. Zhang (✉) · L. Zhang
School of Computer Science and Technology, Anhui University, Hefei 230039, China
e-mail: zhangyp2@gmail.com

C. Xu
Anhui Provincial Hospital, Hefei 230001, China
e-mail: 443779952@qq.com

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_8,

Keywords Quotient space · Granular computing · Equivalence relation · The shortest path

1 Introduction

When we confront complex problems which are hard to handle them accurately, it's not usual to pursue the optimal solution in either systematic or precise way. On the contrary, we reach the limited and reasonable destination step by step, in one way or another, it means we achieve the so-called satisfactory solution. Thanks to the multi-granularity analysis which is sketchy, from coarse to fine and more and more accurate, we successfully avoid the difficulties on the computational complexity. Just in this way, a lot of nonpolynomial questions are smoothly solved.

It is just said by Zhang in [1, 2] "One of the basic characteristics in human problem solving is the ability to conceptualize the world at different granularities and translate from one abstraction level to the others easily, i.e. deal with then hierarchically." Because of the differences of the point of observing an object and the object's further information, a complicated object can be briefed some points that reserve the important characteristics and performances according to the demand to analyse and solve a problem. These points are the representation of different granule worlds.

Existing studies of granular computing typically concentrate on concrete models and computational methods in particular contexts. They unfortunately only reflect specific aspects of granular computing. In fact, there does not exist a formal, precise, commonly agreed, and uncontroversial definition of what is granular computing, nor there is a unified model. Consequently, the potential applicability and usefulness of granular computing are not well perceived and appreciated [3]. Many methods and models of granular computing have been proposed and studied [4–9]. The results enhance our understanding of granular computing. Granular computing comes with a number of interesting pursuits [10–12]. The idea of information granulation offers immediate advantages. It provides tangible benefits in fuzzy modeling by supporting meaningful ways of striking a sound balance between interpretability and accuracy of fuzzy models [13–16], they offer some ways of assessing the performance of the model formed in this way.

The granule world that we define is different from the information granule (IG) that Pawlak proposes. The information granule what is said is a kind of reflection of limited abilities that people deal with and store information, i.e. when facing a lot of complicated information and having the limited abilities, people need partition the information into some simple information blocks according to each characteristic and performance in order to deal with easily. The information block is thought a granule [17–20]. Because information granules are partitioned according to equivalence relation, this only changes the granule of domain of problems and attributed relationship, and the space structure does not been changed. The representation of different granule worlds—quotient space in this paper changes granules not only the domain and attributed relationship but also the space structure of a problem.

In 1990, Bo Zhang and Ling Zhang firstly proposed a new theory, Quotient Space Theory (briefly QST) [1], which was pay high attention to by domestic and overseas scholars. In 1992, the monograph on QST Theory and Applications of Problem Solving [2] was published. In clear–cut classification, we use equivalence relation for establishing our model. A natural question is whether fuzzy equivalence relation can be used for constructing fuzzy classification model. So we have done some research on fuzzy quotient space [21–23]. Recently, we try to use QST to analysis complex networks and some dynamic information [24–27].

When QST contrast with Zadeh's granule computing [28–30], it will transform the original quotient space to fuzzy quotient space with the aid of fuzzy relation of equivalence. Owing to the condition, we think both are similar.

In this paper, we discuss the method how to represent and to partition an object in granular worlds, and educe the relationship of different granular worlds and confirm the degree of granule. We will prove two important theorems of different granules, i.e. to preserve false property theorem and to preserve true property theorem. To solve a problem in different granular worlds, the process procedure of quotient approximate will be applied. We also supply an example of solving problem by different granule worlds—the shortest path of a complex network. The example indicates that to describe or present a complicated object is equal to construct quotient space. In quotient set $[X]$, the complexity to solve a problem is lower than X.

2 Quotient Space

2.1 Basic Definition

A problem solving space, or a problem space for short, is described by a triplet (X, f, Γ). X denotes the problem domain, $f(\cdot)$ indicates the attributes of domain X or is denoted by a function f: $X \to Y$, Γ is the structure of domain X, i.e. the relationship among elements in X. To analyse and solve the triplet (X, f, Γ) of a problem implies analysis and investigation of X, f and Γ.

Assume that X is a domain, R is an equivalence relation on X, $[X]$ is a quotient set under R. Regarding $[X]$ as a new domain, we have a new world which is coarser than X. So a quotient space is a new world that an equivalence relation is thought a new element and coarser than X.

Definition 2.1 A quotient space is the object representation using the quotient set of mathematics to describe or present different granule worlds, i.e. is the method using the quotient set as the mathematic model of different granule worlds.

Problem representations between different granularity sizes correspond to different equivalence relation R or different partitions. So how to partition is the method to construct different granule worlds. We can classify $X = f^{-1}(Y)$ by using the result Y, or classify X directly. In detail there are several methods which supply in next paragraph.

(1) Attribute-based method, namely the same attributions or similar elements are classified.
(2) Projection-based method, consider f is multi-dimensional. Let its n attribute components be $f_1, f_2, \ldots, f_i, f_{i+1}, f_{i+2}, \ldots, f_{n.}$ X is classified with respect to $f_{i+1}, f_{i+2}, \ldots, f_n$ values, while ignoring their attribute components $f_1, f_2, \ldots, f_i$.
(3) Function-based method, a set X of elements is partitioned according to their functions or structures.
(4) Constraint-based method, given n constraints $C_1, C_2, \ldots, C_n$ and a domain X, we may partition X according to C_i, $i = 1, 2, \ldots, n$.

In some cases, some $x \in X$ may belong to more than one class. That is, the classification has overlapped elements or the contour of classes are blurred. We can introduce fuzzy logic for these cases.

Generally, we treat a problem under various grain sizes. Thus, it is necessary to establish the relationship between the worlds at different granularities.

In the book [1, 2] Zhang has discussed the relation between X and $[X]$ and showed that the domains of different granularities are a complete semi-order lattice. But for a problem space (X, f, Γ), structure Γ is very important. When a domain X is discomposed, its structure will change as well. Generally, the coarser the granularities are, the simpler the structure is, however, are there changes of the structure after predigested?

(X, Γ) is a topologic space and Γ is a topology on X. Assume that R is an equivalence relation on X. From R, we have a quotient set $[X]$. A topology $[\Gamma]$ on $[X]$ induced from Γ is called a quotient topology, and $([X], [\Gamma])$ is a quotient topologic space. From topology, it is known that some properties of topologic space (X, Γ) can be observed from its quotient space $([X], [\Gamma])$. We have

Proposition 2.1 *Assume that* p: $(X, \Gamma) \rightarrow ([X], [\Gamma])$ *is a continuous mapping. If* $A \subset X$ *is a connected set on* X, *then* $p(A)$ *is connected set on* $[X]$.

Proposition 2.1 shows that if there is a solution path (connected) in the original domain X, then there exists a solution path in its proper coarse-grained domain $[X]$. Conversely, in the coarse-grained domain, if there does not exist a solution path, there is no solution in the original domain. These properties show that a quotient space has the characteristic of reserving false.

(X, Γ) is a semi-order space or a pseudo semi-order space. In order to establish some relations between semi-order spaces at different grain sizes, we expect to induce a structure $[\Gamma]$ of $[X]$ form Γ of X such that $([X], [\Gamma])$ is also a semi-order space and for all $x, y \in X$, if $x < y$ then $[x] < [y]$. Namely, it is desired that the order relation is preserved invariant in grain size, or order-preserving for short. We can follow the next steps:

(1) transform (X, Γ) into some sort of topologic space.
(2) construct a quotient topologic space $([X], [\Gamma])$ form (X, Γ).
(3) induce a semi-order from $([X], [\Gamma])$ such that the original order relations are preserved in the space.

Proposition 2.2 *Suppose that R is compatible with* Γ. *If* $x, y \in (X, \Gamma)$ *and* $x < y$, *then* $[x] < [y]$, *where* $[x], [y] \in (X, \Gamma)$.

Proof Assume that x is littler than y, define that a is equal to $[x]$.

Assume that $u(a)$ is any opening domain of a on $[X]$, because $p: X \to [X]$ is continuous and $p^{-1}(u(a))$ is not closing on X, thus x is a domain.

Because of $x < y \Rightarrow y \in p^{-1}(u(a)) \Rightarrow p(y) \in u(a) \Rightarrow [y] \in u(a)$, namely $[x] < [y]$. □

Proposition 2.2 indicates that the quotient semi-order space constructed by the preceding approach has order-preserving.

If X is very complicated, we can introduce an equivalence relation and transform X into $[X]$. If R is compatible with Γ, then induces a quotient semi-order $[\Gamma]$ on $[X]$. Thus the old question from x to y is transformed into the new question from $[x]$ to $[y]$. Because R is compatible, then p: $[X, \Gamma] \to ([X], [\Gamma])$ is order preserving. Namely suppose that $[X]$ is a coarse-grained level of X. If there is no solution on some regions of $[X]$, from Proposition 2.2, it is known that there is no solution on the corresponding regions of X as well. Based on the principle, the searching range will be narrowly by pruning off those areas. Since the coarse-grained world usually is simpler than the original one, the searching efficiency will be improved.

Generally, it is not necessary that all characteristics on (X, Γ) are completely mapped onto $([X], [\Gamma])$. This means that in the coarse-grained world some information might be missed due to the abstraction. If the missing characteristics are not interested, that does not matter very much. But the main ones must be preserved in $[X]$.

2.2 To Select Proper *Grain-Size*

Facing different researching goals, there are different quotient sets $[X]$ from the same object (X, f, Γ), and there are different quotient structures $[\Gamma]$ on the same quotient set. Thus how to select proper grain-size is the key to construct reasoning $[X]$, $[\Gamma]$.

In really, the partition is dynamic, namely we partition a X and investigate and research the object and extract some properties on the grain-size firstly, then partition the X again, and so on till the problem is solved. Generally selection and adjustment of grain-size is relation with main domanial acknowledge of the problem. We can select and adjust grain-size by mergence and decomposition.

By merging, we have a new equivalence relation $\underline{R}$ such that (1) $\underline{R} < R$, (2) $\underline{R}$ is compatible, and (3) $\underline{R}$ is maximum. That is, $\underline{R}$ may change R such that the R is compatible and is the coarsest under the supplied condition, namely the number of partition is the least.

By decomposing, we have a new equivalence relation $\overline{R}$ such that (1) $R < \overline{R}$, (2) $\overline{R}$ is compatible, (3) $\overline{R}$ is minimum. That is, $\overline{R}$ may change R such that the R is compatible and is the finest under the supplied condition, namely the number of partition is the biggest.

Thus if R is incompatible with Γ, we can adjust R by merging or decomposing in order that R is compatible, and the compatible result is uniform in condition of maximum or minimum.

The approach we offer for constructing quotient semi-order is the following. First, aright-order topology Γ_R is induced from semi-order Γ. Second, a quotient topology $[\Gamma_R]$ on $[X]$ is induced from Γ_R. Third, a semi-order on $[X]$ is induced from $[\Gamma_R]$. And if R is incompatible with Γ, we can adjust R by merging or decomposing in order that R is compatible.

3 Property Preserving Ability

3.1 Falsity Preserving Principle

We have defined the relation between the domains $[X]$ and X. For a problem space (X, f, T), structure T is very important. When a domain X is decomposed, its structure will change as well. Generally, it is simplified. The main point is whether some properties (or attributes) in X that we are interested in are still preserved after the simplification.

Proposition 3.1 *Assume that R is compatible. If* $x, y \in [x]$ and $x < y$, then interval $[x, y] = \{z | x < z < y, z \in X\} \subset [x]$.

Proof From $x < z < y$ and Proposition 2.2, we have $[x] < [z] < [y]$. Since $[x] = [y] \Rightarrow [x] < [z], [z] < [x]$, from the compatibility of R, we have $[z] = [x] \Rightarrow z \in [x] \Rightarrow [x, y] \subset [x]$. □

Definition 3.1 (X, T) is a semi-order set. $A \subset (X, T)$ is connected $\Leftrightarrow \forall x, y \in$ A, $\exists x = z_1, z_2, \ldots, z_n = y$, such that z_i and z_{i+1}, $i = 1, 2, \ldots, n-1$, are compatible.

Definition 3.2 (X, T) is a semi-order set. $A \subset (X, T)$ is a semi-order closure $\Leftrightarrow$ if $x, y \in A$, and $x < y$, then interval $[x, y] \subset A$.

Corollary 3.1 *In assuming that R is compatible, each component of* $[X]$ *must consist of several semi-order closed, mutually incomparable, and connected sets.*

Note that sets A and B are mutually incomparable, if for $\forall x \in A, y \in B$, x and y are incomparable.

Proof $[X]$ is divided into the union of several connected components, obviously, these components are mutually incomparable. We'll prove that each component is semi-order closed below.

Assume that A is a connected component of $[X]$. If A is not semi-order closed, then $\exists x_1, x_2 \in A$ and $y \notin A$ such that $x_1 < y < x_2$. Since R is compatible, $p : X \to [X]$ is order-preserving. We have $p(x_1) < p(y) < p(x_2) \Rightarrow [x] < [y] < [x] \Rightarrow [y] = [x]$.

That is, $y \in [x]$. Since $y \notin A$, y must belong to another connected component B of $[X]$. Thus, $x_1 \in A$ is comparable with $y \in B$. This contradicts with that components A and B are incomparable. □

Corollary 3.2 *If X is a totally ordered set and R is compatible, then each equivalence class of* $[X]$ *must be an interval $\langle x,y \rangle$, where interval $\langle x,y \rangle$ denotes one of the following four intervals*: $[x, y], [x, y), (x, y], (x, y)$.

Especially, when $x = R^1$ (real number set), Corollary 3.2 still holds.

From the above corollaries, it's known that when partitioning a semi-order set with respect to R, only the corresponding equivalence classes satisfy some structure as shown in above corollaries so that R is compatible. In order to rationally partition a semi-order set, strong constraints have to be followed.

Proposition 3.2 (X, T) *is a semi-order set, then* $((X, T)_r)_s = (X, T)$.

Note: $((X, T)_r)_s$ *is a right semi-order set.*

From the previous discussion, it concludes that a quotient semi-order set ($[X]$, [T]) can be induced from a semi-order set (X, T) so long as R is compatible. And ($[X]$, $[T]$) has order-preserving ability.

When X is a finite semi-order set, it can be represented by a directed acyclic network G. And $x < y \Leftrightarrow$ there exists a directed path in G from x to y. When X is a finite set, X can be represented by a spatial network. We present a simple method for constructing a quotient (pseudo) semi-order on $[X]$ below.

Given (X, T) and an equivalence relation R, we have a quotient set $[X]$. Define a relation "<" on $[X]$ as $\forall a, b \in [X], \exists x_1 \in a, x_2 \in b, x_1 < x_2 \Rightarrow a < b$. Finding the transitive closure of relation "<", have a pseudo semi-order $[T]'$ and quotient space ($[X]$, $[T]'$).

Proposition 3.3 $[T]' = [T]_r]_s$ *holds, where* $[T]'$ *as defined above.*

Using Proposition 3.3, when X is a finite set, the directed graph corresponding to (pseudo) semi-order on $[X]$ can easily be defined as follows: $\forall a, b \in [X], a \to b \Leftrightarrow \exists x, y \in X, x \in a, y \in b, x < y$, where $a \to b$ means there exists a directed edge between a and b. The (pseudo) semi-order corresponding to the directed graph is just the quotient structure on $[X]$.

3.2 Falsity (Truth) Preserving Principle

The order-preserving ability among different grain-size worlds has an extensive application. For example, the relation among elements of domain X is represented by some semi-order structure. A starting point $x \in X$ is regarded as a premise and a goal point $y \in X$ as a conclusion. Whether the directed path from point x to point y exists corresponds to whether conclusion y can be inferred from premise x. If X is complex, introducing a proper partition R to X, then we have $[X]$. A quotient (pseudo) semi-order $[T]_s$ on $[X]$ can be induced. Due to the following proposition, the original directed path finding from x to y on X is transformed into that from $[x]$ to $[y]$ on $[X]$.

Proposition 3.4 *(X, T) is a semi-order set. R is an equivalence relation on X. For $x, y \in X$, if there exists a directed path from x to y on (X, T), there also exists a directed path from $[x]$ to $[y]$ on $[X]$.*

The proposition shows that if the original problem (domain) in hand is too complex, by a proper partition, the original domain is transformed into a coarse one. If there does not exist a solution in the coarse world, then the original problem does not have a solution as well. Since the coarse world is generally simpler than the original one, the problem solving will be simplified.

Note that in Proposition 2.2, even R is incompatible, the order-preserving ability still holds.

From the previous discussion, it is know that an "inference" can be transformed into a spatial search from a premise to a conclusion, i.e., a path-search in a topologic space. And if an original problem (x,f,T) is too complex, then the problem can be transformed into its quotient space $([X],[f],[T])$ which generally simpler than the original one. The order-preserving and the falsity (truth) preserving ability that we will mention below clarify the main characteristics of the multi-granular world; which provide a theoretical foundation for multi-granular computing (inference).

Theorem 3.1 (Falsity Preserving Principle) *If a problem $[A] \to [B]$ on quotient space $([X],[f],[T])$ has no solution, then problem $A \to B$ on its original space (X,f,T) has no solution either. In other words, if $A \to B$ on (X,f,T) has a solution, then $[A] \to [B]$* on *$([X],[f],[T])$ has a solution as well.*

Proof If problem $A \to B$ has a solution, then A and B belong to the same (path) connected set C of (X,f,T). Let $p : X \to [X]$ be a natural projection. Since p is continuous, $p(C)$ is (path) connected on $([X],[f],[T])$. $p(A) = [A]$ and $p(B) = [B]$ belong to the same (path) connected set of $([X],[f],[T])$. The problem $[A] \to [B]$ has a solution. □

Falsity preserving ability within a multi-granular world is unconditional but truth preserving is conditional.

Theorem 3.2 (Truth Preserving Principle I) *A problem* $[A] \to [B]$ *on* $([X], [f], [T])$ *has a solution, if for* $[x]$, $p^{-1}([x])$ *is a connected set on* X, *problem* $A \to B$ *on* (X, f, T) *has a solution.*

Proof Since problem $[A] \to [B]$ on $([X], [f], [T])$ has a solution, $[A]$ and $[B]$ belong to the same connected component C. Letting $D = p^{-1}(C)$, we prove that D is a connected on X.

Reduction to absurdity: Assume that D is partitioned into the union of mutually disjoint non-empty open close sets D_1 and D_2. For $\forall a \in C$, $p^{-1}(a)$ is connected on X, then $p^{-1}(a)$ only belongs to one of D_1 and D_2. $D_i, i = 1, 2$, composes of elements of $[X]$. There exist C_1, C_2 such that $D_1 = p^{-1}(C_1), D_2 = p^{-1}(C_2)$. Since $D_i, i = 1, 2$, are open close sets on X and p is a natural projection, C_1, C_2 are non-empty open close sets on $[X]$. And C_1 and C_2 are the partition of C, then C is non-connected. This is a contradiction. □

Theorem 3.3 (Truth Preserving Principle II) (X_1, f_1, T_1) *and* (X_2, f_2, T_2) *are two quotient spaces of* (X, f, T). $T_i, i = 1, 2$ *are semi-order.* (X_3, f_3, T_3) *is the supremum space of* (X_1, f_1, T_1) *and* (X_2, f_2, T_2). *If problems* $A_1 \to B_1$ *and* $A_2 \to B_2$ *have a solution on* (X_1, f_1, T_1) *and* (X_2, f_2, T_2), *respectively, then problem* $A_3 \to B_3$ *on* (X_3, f_3, T_3) *have a solution, where* $A_3 = A_1 \cap A_2$, $B_3 = B_1 \cap B_2$.

3.3 Computational Complexity Analysis

Using the falsity and truth preserving principle, the computational cost of the multi-granular computing (or inference) can greatly be reduced. For example, by choosing a proper quotient space and using falsity preserving principle, the part of the space without solution can be removed for further consideration so that the computing is accelerated. Similarly, by choosing a proper quotient space and using the truth preserving principle I, the problem solving on the original space can be simplified to that on its quotient space. In general, the size of the quotient space is much smaller than that of the original one so the computational cost is reduced. Concerning the truth preserving principle II, assume that n and m are the potentials of X_1 and X_2, respectively. The potential of X_3 is nm at most. Let $g(\cdot)$ be the computational complexity. When the problem is solved on X_3 directly, $g(\cdot) = g(nm)$. If using the truth preserving principle II, the problem is solved on X_1 and X_2, separately, the computational complexity is $g(n) + g(m)$. This is equivalent to reducing the complexity from $O(N)$ to $O(\ln N)$.

Note that T_3, a topology on (X_3, f_3, T_3), is not necessarily an induced quotient topology $[T_3]$ of X_3, generally $T_3 < [T_3]$. Namely, (X_3, f_3, T_3) is only an element of complete semi-order lattice $\boldsymbol{V}$ but $\boldsymbol{U}$.

4 The Hierarchical Quotient-Space Model of Complex Networks

We have already set up the model of quotient space, and found the Theorems 3.1–3.3 which indicated the change of the main characters during the procedure of granular computing.

In the era of big data, we are forced to confront complex big data sets. In order to analysis these data sets, we utilize multi granular spaces based on quotient spaces. During the process we may reduce the complexity of the data set and solve these big data sets.

Next we present a hierarchical quotient-space model that reduces the computational complexity. We discuss the model to solve the shortest path in a complex network. A complex network is represented by an undirected weighted graph (X, E), where X is a set of nodes, E is a set of edges, f: $E \to R+$, $f(e) \in [0, w]$ is the weight of edge e. Weight w indicates flux, bandwidth or traffic, etc., that is, the reciprocal of distance. An optimal path is the path that connects any pair of nodes with the maximal weight. Then, a shortest path is the path with minimal distance (the reciprocal of weight). Let a set of weights on edges be $\{w^1 > w^2 > \cdots > w^k\}$. In the following discussion, space (or graph) is denoted by (X, E), or simply by X.

Definition 4.1 Equivalence relation $R(w_i)$ is defined as

$$x \sim y \Leftrightarrow \exists x = x_1, x_1, \ldots, x_m = y, f(x_j, x_{j+}) \geq w_i, j =, \ldots, m - 1, i = 1, \ldots k$$

Define $X_i = \{x_1^i, \ldots, x_{n_i}^i\}$, $i = 1, \ldots, k$ as a quotient space corresponding to $R(w_i)$. Let $X = X_0$, and x_i^0 be the element of X. Ranking the elements (nodes) of quotient space X_i, we have a space denoted by $X_i = \{x_1^i, \ldots, x_{n_i}^i\}$, $i = 1, \ldots, k$, as well. Obviously, $(X_0, X_1, \ldots, X_k)$ forms a sequence of hierarchical quotient spaces. Now, the elements in space X are represented by a hierarchical encoding as follows.

For $z \in X$, z is represented by a $(k + 1)$-dimensional integral $z = (z_0, z_1, \ldots, z_k)$. Assume that $p_i : X \to X_i$ is a natural projection. If $p_i(z) = x_t^i$, let the ith coordinate of z be t, i.e., $z_t = t$. It means that if z belongs to the tth element of X_i, then $z_i = t$.

For space X, define a set of its edges as $E_0 : e = (x_j^0, x_t^0) \in E_0 \Leftrightarrow f(x_j^0, x_t^0) \geq w_1$. Simply, let edge $e(x_j^0, x_t^0) = (x_j^0, x_t^0)$. This way, we construct (X_0, E_0).

For space X_1, define a set of its edges as

$E_1 : (x_j^1, x_t^1) \in E_1 \Leftrightarrow \exists x_j^0, x_t^0 \in X, x_j^0 \in x_j^1, x_t^0 \in x_t^1, f(x_j^0, x_t^0) \geq w_2$. Edge $e(x_j^1, x_t^1)$ is represented by

$$e(x_j^1, x_t^1) = \{((x_j^0, p_1(x_j^0)), (x_t^0, p_1(x_t^0)))|\forall x_j^0, x_t^0 \in X, x_j^0 \in x_j^1, x_t^0 \in x_t^1, f(x_j^0, x_t^0) \geq w_2\}$$

Edge $e(x_j^1, x_t^1)$ in space X_1 is a set of edges in space X denoted by e_{jt}^1.

Generally, for space X_i, define a set of its edges as $E_i : (x_j^i, x_t^i \in E_i) \Leftrightarrow \exists x_j^0 \in x_j^i, x_t^0 \in x_t^i, f(x_j^0, x_t^0) \geq w_{i+1}$. Edge $e(x_j^i, x_t^i)$ is represented by $e(x_j^i, x_t^i) = ((x_j^0, p_1(x_j^0), \ldots, p_i(x_j^0) = x_j^i), (x_t^0, p_1(x_t^0), \ldots, p_i(x_t^0) = x_t^i)) | \forall x_j^0 \in x_j^i, x_t^0 \in x_t^i, f(x_j^0, x_t^0) \geq w_{i+1}\}$ or denoted by e_{jt}^i.

Finally, we construct (X_i, E_i), $i = 0, 1, \ldots, k$, of the quotient spaces.

Definition 4.2 $\forall x, y \in X, w_i$, x and y are called w_i-connected $\Leftrightarrow$ there exists an edge from x to y on space X and its weight is greater than or equal to w_i.

Theorem 4.1 $\forall x = (x_0, x_1, x_2, \ldots, x_k), y = (y_0, y_1, y_2, \ldots, y_k) \in X, w_i$, *x and y are w_i-connected $\Leftrightarrow$ $x_i = y_i$, where $x = (x_0, x_1, \ldots, x_k)$ and $y = (y_0, y_1, \ldots, y_k)$ are the hierarchical codes of x and y, respectively. x_i, and y_i are denoted by the corresponded codes of x and y in the sequence of hierarchical quotient spaces* $(X_0, X_1, \ldots, X_k)$.

Proof Assume $x_i = y_i$. From definition of x_i, and y_i, it's known that x and y belong to the same connected component on space X_i. Then $p_{i-1}(x)$ and $p_{i-1}(y)$are w_i-connected on space (X_{i-1}, E_{i-1}). On the other hand, $p_{i-2}(z), z \in X_{i-2}$ is w_{i-1}-connected on space X_{i-2}. From the "truth preserving" property in quotient space theory [26, 27], $p_{i-2}(x)$ and $p_{i-2}(y)$ are w_i-connected on space X_{i-2}. By using the "truth preserving" property gradually, we have that x and y are w_i-connected on space X.

Contrarily, if x and y are w_i-connected on space X, obviously, we have $p_i(x) = p_i(y)$ on space X_i, i.e., $x_i = y_i$. □

For each element x_m^i in space (X_i, E_i), construct a matrix P_m^i. Assume that element x_m^i is composed of s elements of space X_{i-1}. Construct an $s \times s$ matrix P_m^i as follows:

$$P_m^i(tj) = \begin{cases} e(x_t^{i-1}, x_j^{i-1}),\ (x_t^{i-1}, x_j^{i-1}) \in E_{i-1}, & \text{if } , m = 1, \ldots, n_i; \\ \emptyset, & \text{otherwise.} \end{cases}$$

Thus, the topological structure of space X_i can be represented by a set $\{P_j^i, j = 1, \ldots, m\}$ of matrices.

In conclusion, the procedure for constructing the hierarchical quotient space model of network (X, E) is shown below:

(i) According to equivalence relation $R(w_1)$, the elements (nodes) of a weighted edge graph (X, E) are classified into several equivalence classes. Based on the classification, we have a quotient space (X_1, E_1), $X_1 = \{x_1^1, \ldots, x_{n_1}^1\}$, and its corresponding matrices $P_1^1, \ldots, P_{n_1}^1$.

(ii) According to equivalence relation $R(w_2)$, the elements (nodes) of the quotient space (X_1, E_1) are further classified into several equivalence

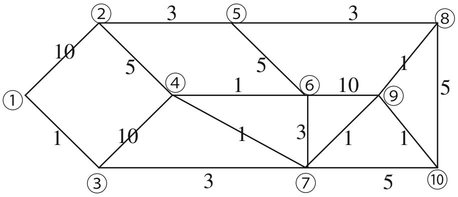


Fig. 1 A weighted network

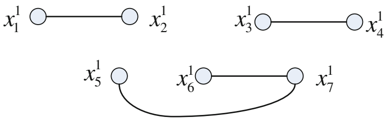


Fig. 2 The quotient space (X_1, E_1)

classes. Then we have a quotient space (X_2, E_2), $X_2 = \{x_1^2, \ldots, x_{n_2}^2\}$, and its corresponding matrices $P_1^2, \ldots, P_{n_2}^2$.

(iii) Generally, according to equivalence relation $R(w_i)$, the elements (nodes) of the space (X_{i-1}, E_{i-1}) are classified into several equivalence classes. We have a quotient space (X_i, E_i), $X_i = \{x_1^i, \ldots, x_{n_i}^i\}$, and its corresponding matrices $P_1^i, \ldots, P_{n_i}^i$, $1 \le i \le k$, where k is the number of different weights on edges.

(iv) The construction of quotient spaces will be ended until space (X_j, E_j), $1 \le j \le k$, has only one element or space (X_k, E_k) is obtained.

(v) Ranking the elements of space X_i, we have a sequence of hierarchical quotient spaces $X_0, X_1, \ldots, X_j$. Each element (node) of space X has a hierarchical code $z = (z_0, z_1, \ldots, z_k)$, $z \in X$. When $j < k$, $z = (z_0, z_1, \ldots, z_j)$, $z \in X$.

Example 4.1 Find the hierarchical quotient-space model of network in Fig. 1.

In Fig. 1, there are 10 nodes {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, and a set $w = \{w_1, w_2, w_3, w_4\} = \{10, 5, 3, 1\}$ of weights.

The given space (X_0, E_0) with 10 elements (nodes):

$$X_0 = \{x_1^0 = (1),\ x_2^0 = (2),\ x_3^0 = (3),\ x_4^0 = (4),\ x_5^0 = (5),\ x_6^0 = (6),\ x_7^0 = (7),\\ x_8^0 = (8),\ x_9^0 = (9),\ x_{10}^0 = (10)\}$$

From equivalence relation $R(10)$, we have a quotient space X_1 with 7 nodes

$$X_1 = \{x_1^1 = (1,2),\ x_2^1 = (3,4),\ x_3^1 = (5),\ x_4^1 = (6,9),\ x_5^1 = (7),\\ x_6^1 = (8),\ x_7^1 = (10)\}.$$

Its corresponding matrices are

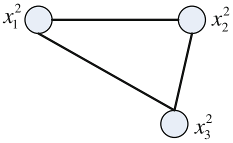


Fig. 3 The quotient space (X_2, E_2)

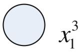


Fig. 4 The quotient space (X_3, E_3)

$$P_1^1 = \begin{pmatrix} 1 & (1,2) \\ & 1 \end{pmatrix},\ P_2^1 = \begin{pmatrix} 1 & (3,4) \\ & 1 \end{pmatrix},\ P_4^1 = \begin{pmatrix} 1 & (6,9) \\ & 1 \end{pmatrix},$$
$$P_3^1 = P_5^1 = P_6^1 = P_7^1 = (1),$$

and the corresponding quotient space (X_1, E_1) as shown in Fig. 2.

From equivalence relation $R(5)$, we have a quotient space X_2 with 3 nodes

$$X_2 = \{x_1^2 = (1,2,3,4),\ x_2^2 = (5,6,9),\ x_3^2 = (7,8,10)\}.$$

Its corresponding matrices

$$P_1^2 = \begin{pmatrix} 1 & ((2,1),(4,2)) \\ & 1 \end{pmatrix}, P_2^2 = \begin{pmatrix} 1 & ((5,3),(6,4)) \\ & 1 \end{pmatrix},$$
$$P_3^2 = \begin{pmatrix} 1 & 0 & ((7,5),(10,7)) \\ & 1 & ((8,6),(10,7)) \\ & & 1 \end{pmatrix},$$

and the corresponding quotient space (X_2, E_2) as shown in Fig. 3.

From equivalence relation $R(3)$, we have a quotient space X_1 with only one node

$x_1^3 = (1,2,3,4,5,6,7,8,9,10)$, its corresponding matrix

$$P_1^3 = \begin{pmatrix} 1 & ((2,1,1),(5,3,2)) & ((3,2,1),(7,5,3)) \\ & 1 & ((5,3,2),(8,6,3))|((6,4,2),(7,5,3)) \\ & & 1 \end{pmatrix},$$

and the corresponding quotient space (X_3, E_3) as shown in Fig. 4.

It's noted that matrix P_j^i is anti-symmetric, i.e., if element $p_{tj}^i = (a,b)$, then $p_{jt}^i = (b,a)$.

Finally, we have the hierarchical codes of each node in space (X_0, E_0) as follows:

$$1 = (1,1,1,1), 2 = (2,1,1,1), 3 = (3,2,1,1), 4 = (4,2,1,1), 5 = (5,3,2,1),$$
$$6 = (6,4,2,1), 7 = (7,5,3,1), 8 = (8,6,3,1), 9 = (9,3,2,1), 10 = (10,7,3,1).$$

5 New Algorithm for Finding Optimal Paths

The optimal path finding procedure begins from the comparison between the last code words in the hierarchical codes of the source node and the destination node to look for the connected path between these two nodes. The procedure carries out from the coarsest quotient space to the finest one gradually until the optimal path is found.

For example, source node $x = (x_0, x_2, \ldots, x_k)$ and destination node $y = (y_0, y_1, y_2, \ldots, y_k)$ in space (X_0, E_0) are given. Compare the last code word x_k with y_k. If $x_k = y_k$, then compare x_{k-1} with y_{k-1} until $x_{i-1} \neq y_{i-1} (0 \leq i \leq k)$ and $x_i = y_i$ so x and y are connected in quotient space (X_{i-1}, E_{i-1}) and equivalent in space (X_i, E_i). Thus, in order to find the connected path between x and y, it's needed to find the connected path between x_{i-1} and y_{i-1} in space (X_{i-1}, E_{i-1}) first. From $P^i_{x_i}$, we may find a connected path $e(x_{i-1}, y_{i-1})$ from x_{i-1} to y_{i-1} in space (X_{i-1}, E_{i-1}). For simplicity, assume that $e(x_{i-1}, y_{i-1}) = (x^1, x^2)$, $x^1 = (x^1_0, \ldots, x^1_{i-1} = x_{i-1})$, and $x^2 = (x^2_0, \ldots, x^2_{i-1} = y_{i-1})$. Inserting x^1 and x^2 into (x, y), we have (x, x^1, x^2, y). Where the $(i - 1)$th coordinates of x and x^1 (or x^2 and y) are the same. For x and x^1, the same operation is implemented, i.e., comparing x_{i-2} with x^1_{i-2} until $x_{j-1} \neq x^1_{j-1}$ and $x_j = x^1_j (0 \neq j < i \leq k)$. Finding the connected path in space (X_{j-1}, E_{j-1}), from $P^j_{x_j}$, it's known that $e(x_{j-1}, x^1_{j-1})$ is the connected path from x to x^1. Insert $e(x_{j-1}, x^1_{j-1})$ into x and x^1. The process carries out until the connected path is found on space (X_0, E_0). For x^2 and y, compare x^2_{i-2} with y_{i-2} until $x^2_{j'-1} \neq y_{j'-1}$ and $x^2_{j'} = y_{j'} (0 \leq j' < i \leq k)$. Finding the connected path in space $(X_{j'-1}, E_{j'-1})$, from $P^{j'}_{x_{j'}}$, we know that $e(x^2_{j'-1}, y_{j'-1})$ is the connected path from x^2 to y. Insert $e(x^2_{j'-1}, y_{j'-1})$ into x^2 and y. The procedure continues until the path is found in space (X_0, E_0).

5.1 The Optimal Path Finding Algorithm

Given $x = (x_1, \ldots, x_k)$ and $y = (y_1, \ldots, y_k)$ in space (X_0, E_0). Assume that $x_i = y_i, x_j \neq y_j$, $j < i$. Find the w_i-connected edge between x and y on space X_0.

Node x is represented by $x = (p_0(x), p_1(x), \ldots, p_k(x))$, where $p_i : X \to X_i$, $i = 0, 1, \ldots, k$, and $X = X_0$. For $x = (x_1, \ldots, x_k)$ and $y = (y_1, \ldots, y_k)$, by assuming that $x_k = y_k$, x and y are connected in space (X_{k-1}, E_{k-1}) and equivalent in space (X_k, E_k).

(i) From $P^k_{x_k}$, in space (X_{k-1}, E_{k-1}) we have a path $e(x_{k-1}, y_{k-1})$ composed of a_k nodes from x_{k-1} to y_{k-1}. Inserting the a_k nodes into (x, y) in turn, we have a sequence composed of $a_k + 2$ nodes. In the sequence, there is a w_k-edge connected the 2*i*th with the $(2i + 1)$th nodes but no edge between the $(2i - 1)$th and the 2*i*th nodes, $i = 1,\ldots, a_k + 1$. Since the $(k - 1)$th coordinates of x and y are the same, the two nodes are connected in space (X_{k-2}, E_{k-2}).

(ii) Let $k \leftarrow k - 1$, go to step (i).

(iii) The procedure continues until the 0th coordinates of the $(2i - 1)$th and the 2*i*th nodes are the same. The sequence of the 0th coordinates is the optimal path.

Example 5.1 Find the optimal path between node 5 and node 7 in Fig. 1.

From Example 4.1, we have a set of quotient spaces as follows: $X_0 = \{x_1^0 = (1),\ x_2^0 = (2),\ x_3^0 = (3),\ x_4^0 = (4),\ x_5^0 = (5),\ x_6^0 = (6),\ x_7^0 = (7), x_8^0 = (8),\ x_9^0 = (9),\ x_{10}^0 = (10)\}X_1 = \{x_1^1 = (1,2),\ x_2^1 = (3,4),\ x_3^1 = (5),\ x_4^1 = (6,9),\ x_5^1 = (7),\ x_6^1 = (8)x_7^1 = (10)\}X_2 = \{x_1^2 = (1,2,3,4,),\ x_2^2 = (5,6,9),\ x_3^2 = (7,8,10)\}.X_3 = \{x_1^3 = (1,2,3,4,5,6,7,8,9,10)\}$.

The hierarchical codes of the source node (node 5) and the destination node (node 7) are $x = (5, 3, 2, 1)$ and $y = (7, 5, 3, 1)$, respectively, i.e., $(x, y) = ((x_0, x_1, x_2, x_3), (y_0, y_1, y_2, y_3)) = ((5, 3, 2, 1), (7, 5, 3, 1))$. By comparing the code words of the hierarchical code of x with that of y, it's known that $x_3 = y_3 = 1$ but $x_2 \neq y_2$. This means that node 5 and node 7 are connected in space (X_2, E_2) and equivalent in space (X_3, E_3), where node 5 and 7 belong to nodes x_2^2 and x_3^2 in space (X_2, E_2), respectively. From matrix P_1^3, we have an w_3-edge between nodes (5, 3, 2) and (8, 6, 3), and an w_3-edge between nodes (6, 4, 2) and (7, 5, 3) in space (X_2, E_2). Inserting these nodes into (x, y), then we have two paths ((5, 3, 2, 1), (5, 3, 2), (8, 6, 3), (7, 5, 3, 1)) and ((5, 3, 2, 1), (6, 4, 2), (7, 5, 3), (7, 5, 3, 1)) from node x to y.

In path ((5, 3, 2, 1), (5, 3, 2), (8, 6, 3), (7, 5, 3, 1)), comparing node (5, 3, 2, 1) with node (5, 3, 2), we have $x_2 = y_2 = 2$, $x_1 = y_1 = 3$, $x_0 = y_0 = 5$. There is a path from node x_1^3 in space (X_3, E_3) to node x_2^2 in space (X_2, E_2). Comparing node (8, 6, 3) with node (7, 5, 3, 1), since $x_2 = y_2 = 3, x_1 \neq y_1$, nodes 8 and 7 belong to nodes x_6^1 and x_5^1 in space (X_1, E_1), respectively. The two nodes are connected in space (X_1, E_1) and equivalent in space (X_2, E_2). From matrix P_3^2, it's known that there is an w_2-edge between nodes (8, 6) and (10, 7) and an w_2-edge between nodes (10, 7) and (7, 5) in space (X_1, E_1). Inserting these nodes into the sequence above, we have a path ((5, 3, 2, 1), (5, 3, 2), (8, 6, 3), (8, 6), (10, 7), (10, 7), (7, 5), (7, 5, 3, 1)). Comparing node (8, 6, 3) with node (8, 6), since $x_1 = y_1 = 6$, $x_0 = y_0 = 8$, there is a path from node x_2^3 in space (X_2, E_2) to node x_6^1 in space (X_1, E_1). Nodes (10, 7) and (10, 7) belong to the same node x_7^1 in space (X_1, E_1). Comparing node (7, 5) with node (7, 5, 3, 1), since $x_1 = y_1 = 5$, $x_0 = y_0 = 7$ there is a path from node x_5^1 in space (X_1, E_1) to node x_1^3 in space (X_3, E_3). Finally, we have an optimal path (5, 8, 10, 7) from node 5 to node 7.

Similarly, from sequence ((5, 3, 2, 1), (6, 4, 2), (7, 5, 3), (7, 5, 3, 1)), we have another optimal path (5, 6, 7) from node 5 to node 7.

The path finding procedure is shown in Fig. 5.

6 Experimental Results

In order to compare our algorithm (Hierarchical Quotient Space Model based Algorithm, HQSM algorithm) with other well-known algorithms, we carried out a set of computer simulations. The experimental environment is a java platform. The undirected and weighted networks are generated by random network, small-world network and scale-free network models, respectively. The edge weights are assigned from [31, 32]. The Dijkstra, Floyd and HQSM algorithms are implemented for finding the optimal path of any pair of nodes in the networks.

6.1 The Shortest Path Quality Comparison

For comparison, we replace the edge weight of networks by its reciprocal. Then, the shortest path finding problem is transformed into that of the optimal path finding with the minimal reciprocal sum. We choose random, small-world, and scale-free networks with 100, 200, 300, 400, and 500 nodes as test beds. The Dijkstra, Floyd and HQSM algorithms are implemented for finding the optimal path of any pair of nodes in the networks. The shortest paths found by Dijkstra and Floyd algorithms are always global minimal. The percentage of the shortest paths found by HQSM algorithm that belong to the global minimum in relation to the total number of the shortest paths found is shown in Tables 1, 2 and 3. The number of hierarchical levels used in the quotient-space approach is also shown in the same tables.

6.2 The Computational Complexity Comparison

The undirected and weighted networks with 100, 200, 300, 400 and 500 nodes involve the three network models, i.e., the random, small-world and scale-free

Table 1 Random networks

Number of nodes	Percentage	Number of levels
100	94.60	6
200	95.50	6
300	97.90	6
400	92.60	6
500	97.90	6

Table 2 Small-world networks

Number of nodes	Percentage	Number of levels
100	90.90	5
200	93.60	5
300	85.90	5
400	93.10	5
500	98.70	4

Table 3 Scale-free networks

Number of nodes	Percentage	Number of levels
100	97.10	9
200	98.20	8
300	98.60	7
400	97.40	7
500	99.26	6

Table 4 Total CPU time (in seconds) in the random network

Number of nodes	100	200	300	400	500
HQSM	0.397	1.356	3.112	6.634	12.171
Dijkstra	1.719	9.797	91.141	656.391	1,002.125
Floyd	0.940	4.220	118.630	212.030	511.560

Table 5 Total CPU time (in seconds) in small-world network

Number of nodes	100	200	300	400	500
HQSM	0.640	2.403	6.659	14.33	22.262
Dijkstra	2.758	10.546	87.239	700.540	1,100.200
Floyd	0.790	4.783	132.743	230.412	498.317

Table 6 Total CPU time (in seconds) in scale-free network

Number of nodes	100	200	300	400	500
HQSM	0.437	1.734	4.297	8.187	15.562
Dijkstra	2.045	8.236	86.431	668.400	998.354
Floyd	0.780	4.060	108.598	206.580	466.250

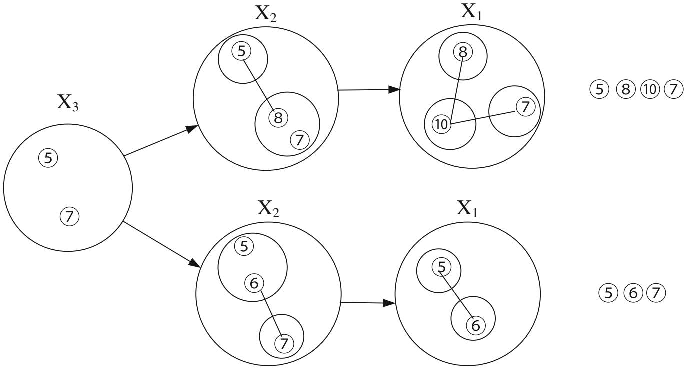


Fig. 5 The optimal paths from node 5 to node 7

networks. The total CPU time for finding the optimal paths of each algorithm is shown in Tables 4, 5 and 6.

The experimental results show that HQSM algorithm outperforms the Dijkstra and Floyd algorithms greatly in saving the computational cost, especially when the networks become larger. On the other hand, Dijkstra and Floyd algorithms need large storage, it's quite difficult to implement the algorithms in networks with more than 500 nodes. In HQSM, the algorithm looks for the optimal paths from the coarse level to the fine one via the hierarchical quotient space model. So it always chooses the high weight edges with high priority. For example, in Fig. 5, space (X_1, E_1) contains $w(5)$-edges (5 → 6, 8 → 10 → 7) and space (X_2, E_2) contains $w(3)$-edges (5 → 8, 6 → 7). HQSM looks for the optimal paths from space (X_2, E_2) to space (X_1, E_1). Then the paths (5 → 6→7, and 5 → 8→10 → 7) are found. But Dijkstra algorithm visits nodes 2, 4, 6, 8 first, then nodes 9, 10, 7, and finally finds the optimal path 5 → 6→7. Floyd algorithm finds the optimal paths between any pair of nodes by searching all nodes and edges throughout the networks. Although HQSM algorithm can only find the quasi-optimal paths generally, the experimental results show that more than ninety percent of the shortest paths found by HQSM algorithm is global minimal.

7 Conclusion

In this paper, we present a quotient space representation method for problem solving. Based on the method, a problem is represented by a triplet (*X*, *f*, *T*). It enables us to describe different structures, attribute functions and operations on a domain. Especially, it offers a tool for depicting different grain-size worlds.

When (*X*, *T*) is a topologic or a semi-order space, we discuss how to construct the quotient topology and quotient (pseudo) semi-order on its corresponding quotient space [*X*] and after the construction what kind of quotient structures that we can have. We prove three important theorems of different granules, i.e. to preserve false property theorem and to preserve true property theorem.

We supply an example of solving problem by different granule worlds—the shortest path of a complex network. The example indicates that to describe or present a complicated object is equal to construct quotient space. In quotient set [*X*], the complexity to solve a problem is lower than *X*. So we have a new solution method to analysis a big data based on the quotient space theory.

References

1. Zhang, B., Zhang, L.: Theory and Applications of Problem Solving. North Holland (1992)
2. Zhang, B., Zhang, L.: Theory and Applications of Problem Solving. Tsinghua University Publisher, Beijing (1990). (in Chinese)
3. Yao, Y.Y.: Perspectives of granular computing. In: Proceedings of 2005 IEEE International Conference on Granular Computing, vol. 1, pp. 85–90 (2005)
4. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Kluwer Academic Publishers, Boston (2002)
5. Dupre, J.: The Discorder of Things, Mataphysical Foundations of the Disunity of Science. Harvard University Press, Cambridge (1993)
6. Foster, C.L.: Algorithms, Abstraction and Implementation: Levels of Detail in Cognitive Science. Academic Press, London (1992)
7. Giunchglia, F., Walsh, T.: A theory of abstraction. Artif. Intell. **56**, 323–390 (1992)
8. Hobbs, J.R.: Granularity. In: Proceedings of the 9th International Joint Conference on Artificial Intelligence, pp. 432–435 (1985)
9. Inuiguchi, M., Hirano, S., Tsumoto, S. (eds.): Rough Set Theory and Granular Computing. Springer, Berlin (2003)
10. Pedrycz, W., Bargiela, A.: An optimization of allocation of information granularity in the interpretation of data structures toward granular fuzzy clustering. IEEE Trans. Syst. Man Cybern. B Cybern. **42**(3), 582–590 (2012)
11. Pedrycz, A., Hirota, K., Pedrycz, W., Dong, F.: Granular representation and granular computing with fuzzy sets. Fuzzy Sets Syst. **203**, 17–32 (2012)
12. Pedrycz, W., Song, M.: Granular fuzzy models: a study in knowledge management in fuzzy modeling. Int. J. Approx. Reasoning **53**(7), 1061–1079 (2012)
13. Pedrycz, W., Bargiela, A.: An optimization of allocation of information granularity in the interpretation of data structures: toward granular fuzzy clustering. IEEE Trans. Syst. Man Cybern. B **42**(3), 582–590 (2012)
14. Pedrycz, W., Homenda, W.: Building the fundamentals of granular computing: a principle of justifiable granularity. Appl. Soft Comput. **13**(10), 4209–4218 (2013)

15. Chen, S.-M., Yang, M.-W., Lee, L.-W., Yang, S.-W.: Fuzzy multiple attributes group decision-making based on ranking interval type-2 fuzzy sets. Expert Syst. Appl. **39**(5), 5295–5308 (2012)
16. Chen, S.M., Randyanto, Y.: A novel similarity measure between intuitionistic fuzzy sets and its applications. Int. J. Pattern Recognit. Artif. Intell. **27**(7), 1350021-1–1350021-34 (2013)
17. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers, Dordrecht (1991)
18. Pawlak, Z.: Rough sets. Int. J. Comput. Inform. Sci. **11**, 341–356 (1982)
19. Pawlak, Z.: Rough Sets, Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers, Dordrecht (1991)
20. Pawlak, Z.: Granularity of knowledge, indiscernibility and rough sets. In: Proceedings of 1998 IEEE International Conference on Fuzzy Systems, pp. 106–110 (1998)
21. Zhang, L., Zhang, B.: Quotient Space Theory and Granule computing. CRSSC'2003, Chongqing, China, pp. 1–3 (2003) (in Chinese)
22. Zhang, L., Zhang, B.: Theory of fuzzy quotient space (methods of fuzzy granular computing). J. Softw. **14**(4), 770–776 (2003). (in Chinese)
23. Zhang, Y., Zhang, L., Wu, T.: The representation of different granular worlds: a quotient space. Chin. J. Comput. **27**(3), 328–333 (2004). (in Chinese)
24. Zhang, L., Zhang, B.: The analysis of system performances based on quotient space granular computing. J. Comput. Sci. **31**(10A), 6–9 (2004). (in Chinese)
25. Cheng, W., Shi, Y., Zhang, Y.: Application of quotient space theory in yield prediction. J. Comput. Eng. Appl. **43**(13), 197–199 (2007). (in Chinese)
26. He, F., Zhang, Y., Zhao, S.: The method for the optimal path of complex network based on quotient space hierarchy model (MOCQ). In: Science Paper Online (2007). http://www.paper.edu.cn/releasepaper/content/200712-610 (in Chinese)
27. He, F., Zhang, Y., Chen, J., Zhao, S.: Dynamic information analysis model based on quotient space topology. In: Science Paper Online, vol 5(2), pp. 124–127 (2010) (in Chinese)
28. Zadeh, L.A.: Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets Syst. **19**, 111–127 (1997)
29. Zadeh, L.A.: Fuzzy sets and information granulation. In: Gupta, M., Ragade, R.K., Yager, R.R. (eds.) Advances in Fuzzy Set Theory and Applications. North-Holland Publishing Company, Amsterdam (1979)
30. Zadeh, L.A.: Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. Soft. Comput. **2**, 23–25 (1998)
31. Larson, R., Odoni, A.: Shortest paths between all pairs of nodes. In: Urban Operations Research (1981)
32. Pemmaraju, S., Skiena, S.: All-pairs shortest paths. In: Computational Discrete Mathematics: Combinatorics and Graph Theory in Mathematica, pp 330–331. Cambridge University Press, Cambridge (2003)

Towards an Optimal Task-Driven Information Granulation

Alexander Ryjov

Abstract In this work we have analyzed Big Data sources and made a conclusion that sizeable part of them is people-generated data. We can present this type of data in form of qualitative attributes. The model of such attributes is a collection of fuzzy granules. We also need to granulate the data for application of a big part of analytical technologies. When we form the granules, we have a choice among different variants. Which of them is good for specific task? How can we measure this “goodness” and make a choice the best (optimal) granulation? We provide our vision of answers on these questions in the chapter.

Keywords Big data · Fuzzy information granulation · Fuzzy linguistic scales · Measure of fuzziness · Loss of information and information noise for fuzzy data

1 Motivation

Big Data is recognizable trend in modern Information and Communications Technologies (ICT) with visible impact on different areas of business, economics, politics, and other aspects of society [1–6]. Due to huge scale, features of data source, scenarios of usage, etc., we often cannot use a standard approaches for data processing. In this situation, one of possible ways is using a generalized data representation in form of granules.

When we form these granules, we have a choice among different variants. Which of them is good for specific task? How can we measure this “goodness” and make a choice the best (optimal) granulation? What does this choice mean in terms of

A. Ryjov (✉)
Mathematical Foundations of Intelligent Systems, Department of Mechanics
and Mathematics, Lomonosov’ Moscow State University, Moscow 119899, Russia
e-mail: alexander.ryjov@gmail.com

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_9,

quality of task's solution? In proposed chapter, we will answer on these questions on example on information retrieval task and generalization of this approach.

2 Big Data and Information Granulation

The introduction of the term «big data» refers to Clifford Lynch, editor of the journal Nature, who prepared to 3 September 2008, the special issue of the magazine [7] which examines what big data sets mean for contemporary science, where has collected materials about the phenomenon of explosive growth in the volume and diversity of data and technological prospects in the paradigm of the likely leap «from quantity to quality».

Despite the fact that the term was introduced in the academic environment, first of all, was the problem of growth and diversity of scientific data, as of 2009, the term is widely spread in the business press, and by 2010 to include the first appearance of products and solutions relating solely and directly to the problem of handling large data. By 2011, the majority of the largest providers of information technology for the organizations in their business strategies are based on the concept of large data, including IBM [3], Oracle [4], Microsoft [8], Hewlett-Packard [9], EMC [2], and the basic analytics of the market of information technologies devote the concept of dedicated research [1, 2].

The area "Big Data" is still not well-defined for now; this area looks like "point of gravity" for business and technologies. Analysis of mentioned above sources, the press (for example, [10–13]), and the blogs (for example, [14, 9, 15]) allow us to mark out the following focuses of the discussions:

- Source of big data
- Hardware/infrastructure for big data
- Basic software
- Information technologies/methods/application software
- Usage of big data/business value.

This splitting is not aspiring to completeness (for example, we do not mention a very important ethical implications [16]), but this rough segmentation is enough for our purposes.

We do not have precise information about impact of different sources in generation of data volume. In very general terms we can mark out devices and people as sources of big data. Examples of the first sources are: national and international projects such as the Large Hadron Collider (LHC) at CERN, Europe's particle-physics laboratory near Geneva in Switzerland, or the Large Synoptic Survey Telescope at northern Chile; Internet of Things; industry (SCADA, finance, etc.). Examples of the second type of sources are: social networks, health care, retail, personal location data, public sector administration, etc. Many authors (for example, [1, 9]) note the importance of people-generated data. We will focus on this type of data below.

Many analytical technologies like data mining, decision trees, etc., are working with discrete data sets. It means that we have to understand how we can make a discretization, or granulation, of initial data. For human-created data (in opposite of system-output data—see above) it means a fuzzy information granulation. The founder of fuzzy logic, professor L.A. Zadeh, underlined, that "In coming years, theory of fuzzy information granulation is likely to play an important role in the evolution of fuzzy logic and, in conjunction with computing with words, may eventually have a far-reaching impact on its applications" [17]. Information granularity, representation and design of information granules, and applications in analysis and design of intelligent systems are presented in [18]. Here, we will show how we can make an optimal fuzzy information granulation for different tasks associated with Big Data.

Thus, we can make the following summary of this section:

- Human-created data is a sizable part of "Big Data"
- For application of a big part of analytical technologies we need to granulate the data
- For human-created data it means fuzzy granulation technique

At the end of this section, I would like to note, that author of "Big Data" term, Clifford Lynch, in his editorial paper in 2008 [19] made a summary: "Above all, data on today's scales require scientific and computational intelligence". This book is an attempt to introduce computational intelligence into Big Data area.

3 Optimal Granulation for People-Generated Data

3.1 Problem Statement

It is assumed that the person describes the properties of real objects in the form of linguistic values. The subjective degree of convenience of such a description depends on the selection and the composition of such linguistic values. Let us explain this on a model example.

Example 1 Let it be required to evaluate the height of a man. Let us consider two extreme situations.

Situation 1. It is permitted to use only two values: "*small*" and "*high*".
Situation 2. It is permitted to use many values: "*very small*", "*not very high*", …, "*not small and not high*", …, "*very high*".

Situation 1 is inconvenient. In fact, for many men both the permitted values may be unsuitable and, in describing them, we select between two "bad" values.
Situation 2 is also inconvenient. In fact, in describing height of men, several of the permitted values may be suitable. We again experience a problem but now due to the fact that we are forced to select between two or more "good" values.

The same situation we have when we describe different objects, for example in social networks: restaurants (expensive-cheap; cozy–bleak; delicious–insipid, etc.), music, theaters, etc. One object may be described by different persons. Therefore it is desirable to have assurances that the different participants of the networks describe one and the same object in the most “uniform” way. Could a set of linguistic values be optimal in this case?

This problem may be reformulated as a problem of construction of an optimal information granulation procedure from point of view of criterion 1 and criterion 2.

On the basis of the above we may formulate the first problem as follows:

Problem 1 Is it possible, taking into account certain features of the man’s perception of objects of the real world and their description, to formulate a rule for selection of the optimum set of values of characteristics (collection of fuzzy granules) on the basis of which these objects may be described? Two optimality criteria are possible:

Criterion 1. We regard as optimum those sets of values through whose use man experiences the minimum uncertainty in describing objects.
Criterion 2. If the object is described by a certain number of users, then we regard as optimum those sets of values which provide the minimum degree of divergence of the descriptions.

3.2 Formalization

The model of an estimating of real object’s properties by a person as the procedure of measuring in fuzzy linguistic scale (FLS) has been analyzed at first time in [20] and described in details in [21]. The set of scale values of some FLS is a collection of fuzzy granules defined on the same universum.

Let us consider t fuzzy granules with the names $a_1, a_2, \ldots, a_t$, specified in one universal set (Fig. 1). We shall call such set the scale values set of a FLS.

Let us introduce a system of limitations for the membership functions of the fuzzy variables comprising s_t. For the sake of simplicity, we shall designate the membership function a_j as μ_j. We shall consider that:

1. $\forall \mu_j (1 \leq j \leq t) \exists U_j^1 \neq \emptyset$, where $U_j^1 = \{u \in U : \mu_j(u) = 1\}$, U_j^1 is an interval or a point;
2. $\forall j (1 \leq j \leq t) \mu_j$ does not decrease on the left of U_j^1 and does not increase on the right of U_j^1 (since, according to 1, U_j^1 is an interval or a point, the concepts “on the left” and “on the right” are determined unambiguously).

Requirements 1 and 2 are quite natural for membership functions of granules forming a scale values set of a FLS. In fact, the first one signifies that, for any concept used in the universal set, there exists at least one object which is standard

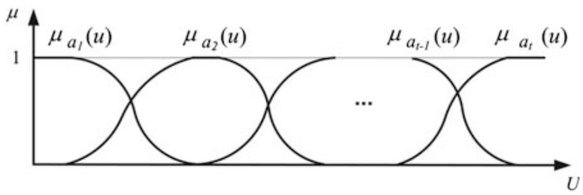


Fig. 1 The scale values set of a FLS

for the given concept. If there are many such standards, they are positioned in a series and are not "scattered" around the universe. The second requirement signifies that, if the objects are "similar" in the metrics sense in a universal set, they are also "similar" in the sense of FLS.

Henceforth, we shall need to use the characteristic functions as well as the membership functions, and so we shall need to fulfil the following technical condition:

3. $\forall j (1 \leq j \leq t) \mu_j$ has not more than two points of discontinuity of the first kind.

For simplicity, let us designate the requirements 1–3 as *L*.

Let us also introduce a system of limitations for the sets of membership functions of fuzzy variables comprising s_t. Thus, we may consider that:

4. $$\forall u \in U \, \exists j (1 \leq j \leq t) : \mu_j(u) > 0;$$

5. $$\forall u \in U \sum_{j=1}^{t} \mu_j(u) = 1.$$

Requirements 4 and 5 also have quite a natural interpretation. Requirement 4, designated the *completeness* requirement, signifies that for any object from the universal set there exists at least one concept of FLS to which it may belong. This means that in our scale values set there are no "holes". Requirement 5, designated the *orthogonally* requirement, signifies that we do not permit the use of semantically similar concepts or synonyms, and we require sufficient distinction of the concepts used. Note also that this requirements is often fulfilled or not fulfilled depending on the method used for constructing the membership functions of the concepts forming the scale values set of a FLS 24.

For simplicity, we shall designate requirements 4 and 5 as *G*.

We shall term the FLS with scale values set consisting of fuzzy variables, the membership functions of which satisfy the requirements 1–3, and their populations the requirements 4 and 5, a *complete orthogonal FLS* and denote it *G(L)*.

As can be seen from example 4, the different FLS have a different degree of internal uncertainty. Is it possible to measure this degree of uncertainty? For complete orthogonal FLS the answer to this question is yes.

3.3 Properties

To prove this fact and derive a corresponding formula, we need to introduce a series of additional concepts.

Let there be a certain population of t membership functions $s_t \in G(L)$. Let $s_t = \{\mu_1, \mu_2, \ldots, \mu_t\}$. Let us designate the population of t characteristic functions $\widehat{s}_t = \{h_1, h_2, \ldots, h_t\}$ as *the most similar population of characteristic functions*, if $\forall j (1 \le j \le t)$

$$h_j(u) = \begin{cases} 1, & \text{if } \mu_j(u) = \max_{1 \le i \le t} \mu_i(u) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

It is not difficult to see that, if the complete orthogonal FLS consists not of membership functions but of characteristic functions, then no uncertainty will arise when describing objects in it. The expert unambiguously chooses the term a_j, if the object is in the corresponding region of the universal set. Some experts describe one and the same object with one and the same term. This situation may be illustrated as follows. Let us assume that we have scales of a certain accuracy and we have the opportunity to weigh a certain material. Moreover, we have agreed that, if the weight of the material falls within a certain range, it belongs to one of the categories. Then we shall have the situation accurately described. The problem lies in the fact that for our task there are no such scales nor do we have the opportunity to weigh on them the objects of interest to us.

However we can assume that, of the two FLS, the one having the least uncertainty will be that which is most "similar" to the space consisting of the populations of characteristic functions. In mathematics, distance can be a degree of similarity. Is it possible to introduce distance among FLS? For complete orthogonal FLS it is possible.

First of all, note that the set of functions L is a subset of integrable functions on an interval, so we can enter the distance in L, for example,

$$\rho(f.g) = \int_U |f(u) - g(u)| du,\ f \in L, g \in L.$$

Lemma 1 *Let* $s_t \in G(L)$, $s'_t \in G(L)$; $s_t = \{\mu_1(u), \mu_2(u), \ldots, \mu_t(u)\}$, $s'_t = \{\mu'_1(u), \mu'_2(u), \ldots, \mu'_t(u)\}$, $\rho(f \cdot g)$*-some distance in L. Then* $d(s_t, s'_t) = \sum_{j=1}^{t} \rho\left(\mu_j, \mu'_j\right)$ *is a distance in G(L).*

The semantic statements formulated by us above may be formalized as follows.

Let $s_t \in G(L)$. For the measure of uncertainty of s_t we shall take the value of the functional $\xi(s_t)$, determined by the elements of *G(L)* and assuming the values in [0,1] (i.e. $\xi(s_t) : G(L) \to [0, 1]$), satisfying the following conditions (axioms):

A1. $\xi(s_t) = 0$, if s_t is a set of characteristic functions;
A2. Let $s_t, s'_{t'} \in G(L)$, t and t' may be equal or not equal to each other. Then $\xi(s_t) \le \xi\left(s'_{t'}\right)$, if $d(s_t, \hat{s}_t) \le d\left(s'_{t'}, \hat{s}'_{t'}\right)$.

(Let us recall that $\hat{s}_t$ is the set of characteristic functions determined by (1) closest to s_t).

Do such functional exist? The answer to this question is given by the following theorem [22].

Theorem 1 (theorem of existence) Let $s_t \in G(L)$. Then the functional

$$\xi(s_t) = \frac{1}{|U|}\int\limits_U f\left(\mu_{i_1^*}(u) - \mu_{i_2^*}(u)\right)du \tag{2}$$

where

$$\mu_{i_1^*}(u) = \max_{1 \le j \le t} \mu_j(u), \quad \mu_{i_2^*}(u) = \max_{1 \le j \le t, j \ne i_1^*} \mu_j(u), \tag{3}$$

f satisfies the following conditions:

F1. f(0) = 1, f (1) = 0;
F2. f does not increase—is a measure of uncertainty of s_t, i.e. satisfies the axioms A1 and A2.

There are many functional satisfying the conditions of Theorem 1. They are described in sufficient detail in 24. The simplest of them is the functional in which the function f is linear. It is not difficult to see that conditions F1 and F2 are satisfied by the sole linear function $f(x) = 1 - x$. Substituting it in (2), we obtain the following simplest measure of uncertainty of the complete orthogonal FLS:

$$\xi(s_t) = \frac{1}{|U|}\int\limits_U \left(1 - \left(\mu_{i_1^*}(u) - \mu_{i_2^*}(u)\right)\right)du, \tag{4}$$

where $\mu_{i_1^*}(u), \mu_{i_2^*}(u)$ are determined by the relations (3).

3.3.1 Interpretation

We can provide the following interpretation of (4). We consider the process of description by person of a real objects. We do not have any uncertainty in the process of a linguistic description of an object which possessing a “physical” significance of the attribute u_1 (Fig. 2).

We attribute it to term a_1 with total reliance. We can to repeat these statement about an object which have “physical” significance of attribute u_5. We choose the

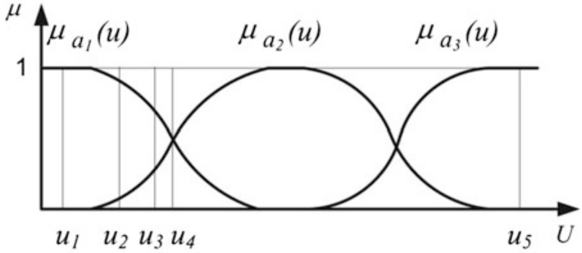


Fig. 2 Interpretation of degree of fuzziness of a FLS

term a_3 for its description without fluctuations. We begin to test the difficulties of choosing of a suitable linguistic significance in the description of an object, possessing the physical significance of attribute u_2. These difficulties grow (u_3) and reach the maximal significance for an objects, possessing the physical significance of attribute u_4: for such objects both linguistic significances (a_1 and a_2) are equal. If we consider the significance of the integrand function

$$\eta(s_t u) = 1 - \left(\mu_{i_1^*}(u) - \mu_{i_2^*}(u)\right)$$

in these points, we can say, that

$$0 = \eta(s_t, u_5) = \eta(s_t, u_1) < \eta(s_t, u_2) < \eta(s_t, u_3) < \eta(s_t, u_4) = 1.$$

Thus, the value of the integral (4) is possible to be interpret as an average human doubts degree while describing some real object.

Note 1. Actually, above we have tacitly assumed that our objects are uniformly distributed in universum. For many real tasks it is not true. To having an average human doubts degree while describing some real object, we have to use a frequency function (if we know one) in sub-integral function. In this case we have the following generalization of (4):

$$\xi(s_t) = \int_U p(u)\left(1 - \left(\mu_{i_1^*}(u) - \mu_{i_2^*}(u)\right)\right)du \tag{5}$$

Below we will analyze the simplest case—uniform distribution.

It is also proved that the functional (4) has natural and good properties for fuzziness degree. In particular the following theorems then hold [22].

Let us define the following subset of function set L:

$\bar{L}$ is a set of functions from L, which are part-linear and linear on $\bar{U} = \left\{u \in U : \forall j (1 \le j \le t)\, 0 < \mu_j(u) < 1\right\}$;

$\hat{L}$ is a set of functions from L, which are part-linear on U (including $\bar{U}$).

Theorem 2 *Let* $s_t \in G(\bar{L})$. *Then* $\xi(s_t) = \frac{d}{2|U|}$, *where* $\mathrm{d} = |\bar{U}|$.

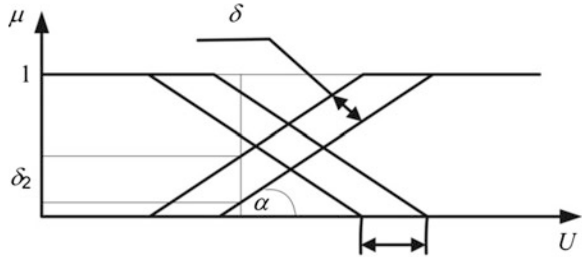


Fig. 3 Presentation of $G^{\delta}(L)$

Theorem 3 *Let* $s_t \in G(\hat{L})$. *Then* $\xi(s_t) = c\frac{d}{|U|}$, *where* $c < 1$, *c = Const.*

Finally, we present the results of the analysis of our model, when the membership functions which are members of the given collection of fuzzy sets, are not given with absolute precision, but with some maximal inaccuracy δ (Fig. 3). Let us call this particular situation the δ-model and denote it by $G^{\delta}(L)$.

In this situation we can calculate the top ($\bar{\xi}(s_t)$) and the bottom ($\underline{\xi}(s_t)$) valuations of the degree of fuzziness.

FLS with minimal and maximal degrees of fuzziness for simple case $t = 2$ is shown on Fig. 4.

The following theorem is hold [22].

Theorem 5 *Let* $s_t \in G^{\delta}(\bar{L})$. *Then* $\underline{\xi}(s_t) = \frac{d(1-\delta_2)^2}{2|U|}$, $\bar{\xi}(s_t) = \frac{d(1+2\delta_2)}{2|U|}c$, *where* $d = |\bar{U}|$.

By comparing the results of the theorem 2 and theorem 5, we see that for small significances δ, the main laws of our model are preserved. Therefore, we can use our technique of estimation of the degree of fuzziness in practical tasks, since we have shown it to be stable.

Based on these results we can propose the following method for optimal information granulation.

3.4 Method for Optimal Granulation for Human-Created Data

1. All the “reasonable” sets of linguistic values are formulated;
2. Each of such sets is represented in the form of $G(L)$;
3. For each set the measure of uncertainty (5) is calculated;
4. As the optimum set minimizing both the uncertainty in the description of objects and the degree of divergence of opinions of users we select the one, the uncertainty of which is minimal.

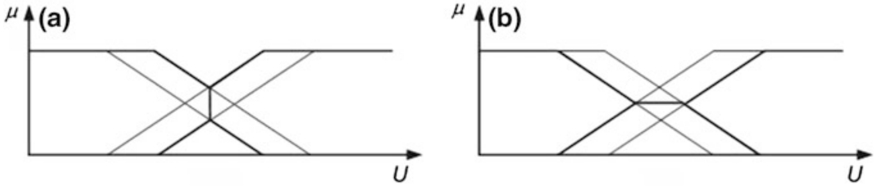


Fig. 4 FLS with minimal (**a**) and maximal (**b**) degrees of fuzziness

Following this method, we may describe objects with minimum possible uncertainty, i.e. guarantee that different people will describe the objects in the most possible unified manner (see Criterion 2 in Problem 1). It means that the number of situations when one real object has more than one image in our Big Data model (for example, in social networks), or different real objects have the same image, will be minimal. Accordingly, we will have a maximal possible adequacy of the data as a model of real world from this point of view. Stability of the measure of uncertainty (Theorem 5) allows us to use this method in practical applications.

Notice, that for different distributions of the objects in real world, we could have a different optimal collection of granules. For example, different countries have a different age structure of the population. It means, that optimal information granulation of age could be different for different countries.

4 Optimal Granulation for Information Retrieval

4.1 Problems Statement

In the chapter, we will analyze the following picture (Fig. 5). Data base is an information model of subject area for the user. The quality of this model depends on the quality of objects' description provided by source of information.

In this context we can formulate the following problem [23].

Problem 2 Is it possible to define the indices of quality of information retrieval in fuzzy (linguistic) databases and to formulate a rule for the selection of such a set of linguistic values, use of which would provide the maximum indices of quality of information retrieval?

This problem may be reformulated as a problem of construction of an optimal information granulation procedure from point of view of information retrieval.

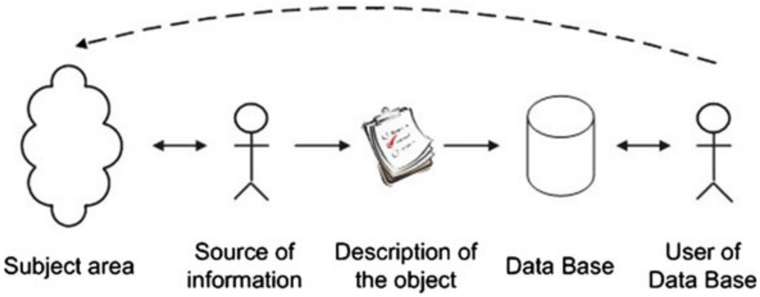


Fig. 5 Data base as a model of subject area

4.2 Formalization

Data base is an information model of the real world (Fig. 5). The quality of this model is expressed, in particular, through parameters of the information retrieval. If the database containing the linguistic descriptions of objects of a subject area allows to carry out qualitative and effective search of the relevant information then the system will work also qualitatively and effectively.

As well as in Sect. 3.2, we shall consider that the set of the linguistic meanings can be submitted as $G(L)$.

In our study of the process of information searches in data bases whose objects have a linguistic description, we introduced the concepts of loss of information ($\Pi_X(U)$) and of information noise ($\mathrm{H}_X(U)$). These concepts apply to information searches in these data bases, whose attributes have a set of significances X, which are modelled by the fuzzy franules in s_t. The meaning of these concepts can informally be described as follows. While interacting with the system, a user formulates his query for objects satisfying certain linguistic characteristics, and gets an answer according to his search request. If he knows the real (not the linguistic) values of the characteristics, he would probably delete some of the objects returned by the system (information noise), and he would probably add some others from the data base, not returned by the system (information losses). Information noise and information losses have their origin in the fuzziness of the linguistic descriptions of the characteristics.

These concepts can be formalized as follows.

Let's consider the case $t = 2$ (Fig. 6).

Let's fix the number $u^* \in U$ and introduce following denotes:

- $N(u^*)$ is the number of objects, the descriptions of which are stored in the data base, that possess a real (physical, not linguistic) significance equal to u^*;
- $N^{\ user}$—the number of users of the system.

Then

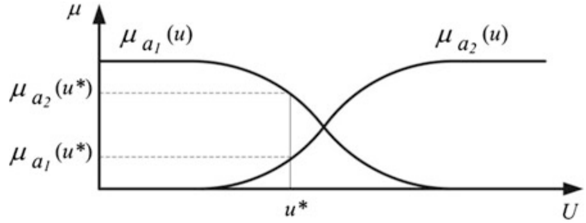


Fig. 6 Simple case $t = 2$

- $N_{a_1}(u^*) = \mu_{a_1}(u^*)N(u^*)$—the number of data base descriptions, which have real meaning of some characteristic equal u^* and is described by source of information as a_1;
- $N_{a_2}(u^*) = \mu_{a_2}(u^*)N(u^*)$—the number of the objects, which are described as a_2;
- $N_{a_1}^{user}(u^*) = \mu_{a_1}(u^*)N^{user}$—the number of the system's users who believe that u^* is a_1;
- $N_{a_2}^{user}(u^*) = \mu_{a_2}(u^*)N^{user}$—the number of the users who believe that u^* is a_2.

That's why under the request "To find all objects which have a meaning of an attribute, equal a_1" (let's designate it as $\langle I(O) = a_1 \rangle$) the user gets $N_{a_1}(u^*)$ descriptions of objects with real meaning of search characteristic is equal to u^*. Under these circumstances $N_{a_1}^{user}(u^*)$ users do not get $N_{a_2}(u^*)$ object descriptions (they carry loses). It goes about descriptions of objects which have the meaning of characteristic equal u^*, but described by sources as a_2. By analogy the rest $N_{a_2}^{user}(u^*)$ users get noise ("unnecessary" descriptions in the volume of given $N_{a_1}(u^*)$ descriptions).

Average individual loses for users in the point $u*$ under the request are equal

$$\pi_{a_1}(u^*) = \frac{1}{N^{user}} N_{a_1}^{user}(u^*) \times N_{a_2}(u^*) = \mu_{a_1}(u^*)\mu_{a_2}(u^*)N(u^*) \tag{6}$$

By analogy average individual noises in the point $u*$

$$h_{a_1}(u^*) = \frac{1}{N^{user}} N_{a_2}^{user}(u^*) \times N_{a_1}(u^*) = \mu_{a_1}(u^*)\mu_{a_2}(u^*)N(u^*) \tag{7}$$

Average individual information loses and noises, given under analyzed request ($\Pi_{a_1}(U)$ and $\mathrm{H}_{a_1}(U)$ accordingly) are naturally defined as

$$\Pi_{a_1}(U) = \frac{1}{|U|}\int_U \pi_{a_1}(u)du, \mathrm{H}_{a_1}(U) = \frac{1}{|U|}\int_U h_{a_1}(u)du$$

It’s obvious that

$$\varPi_{a_1}(u^*) = \mathrm{H}_{a_1}(u^*) = \frac{1}{|U|}\int\limits_U \mu_{a_1}(u)\mu_{a_2}(u)N(u)du \tag{8}$$

By analogy for the request $\langle I(O) = a_2\rangle$ or from symmetry considerations we can get that in this case average loses and noises are equal ($\varPi_{a_2}(U) = \mathrm{H}_{a_2}(U)$) too and are equal the right part of (8). Under information loses and noises appearing during some actions with characteristic which has the set of significance $X = \{a_1, a_2\}$ (($\varPi_X(U)$ and $\mathrm{H}_X(U)$) we naturally understand

$$\varPi_X(U) = p_1\varPi_{a_1}(U) + p_2\varPi_{a_2}(U), \mathrm{H}_X(U) = p_1\mathrm{H}_{a_1}(U) + p_2\mathrm{H}_{a_2}(U),$$

where p_i (i = 1, 2)—the probability of some request offering in some i-meaning of the characteristic.

It’s obvious that as $p_1 + p_2 = 1$, then

$$\varPi_X(U) = \mathrm{H}_X(U) = \frac{1}{|U|}\int\limits_U \mu_{a_1}(u)\mu_{a_2}(u)N(u)du \tag{9}$$

Let’s consider general case: t–meanings of the retrieval attribute. We can generalize the formula (9) in case of t meanings of the retrieval attribute the following way [24]:

$$\varPi_X(U) = \mathrm{H}_X(U) = \frac{1}{|U|}\sum_{j=1}^{t-1}(p_j + p_{j+1})\int\limits_U \mu_{a_j}(u)\mu_{a_{j+1}}(u)N(u)du, \tag{10}$$

where $X = \{a_1, \ldots, a_t\}$, p_i (i = 1, 2, …, t)—the probability of some request offering in some i-meaning of the characteristic.

4.3 Properties

The following theorem is hold [24].

Theorem 6 *Let* $s_t \in G\ (\bar{L})$, *N(u) = N = Const and* $p_j = \frac{1}{t}$ *(j = 1, …, t). Then*

$$\varPi_X(U) = \mathrm{H}_X(U) = \frac{ND}{3t|U|}, \quad \text{where } D = |\bar{U}|$$

Corollary 1 *Let the restrictions of the theorem 6 are true. Then*

$$\Pi_X(U) = \mathrm{H}_X(U) = \frac{2N}{3t}\xi(s_t).$$

For proof of the Corollary is enough to compare theorems 2 and 6.

We can generalize corollary 1 for $s_t \in G(L)$. The following theorem is hold.

Theorem 7 *Let $s_t \in G(L)$, $N(u) = N = Const$ and $p_j = \frac{1}{t}$ $(j = 1, \ldots, t)$. Then* $\Pi_X(U) = \mathrm{H}_X(U) = \frac{c}{t}\xi(s_t)$, where c is a constant with depends from N only.

This theorems showing that the volumes of losses of the information and of information noise arising by search of the information in a data base are coordinated with a degree of uncertainty of the description of objects. It means that describing objects by an optimum way (with minimization of degree of uncertainty) we provide also optimum search of the information in data bases.

By analogue with Sect. 3.3, we can construct the top $(\overline{\Pi}_X(U), \overline{\mathrm{H}}_X(U))$ and bottom $(\underline{\Pi}_X(U), \underline{\mathrm{H}}_X(U))$ valuations of the $\Pi_X(U)$ and $\mathrm{H}_X(U)$.

The following theorems and corollaries are hold [24].

Theorem 8 *Let $X = \{a_1, \ldots, a_t\}$, $s_t \in G^{\delta}(\bar{L})$, $N(u) = N = Const$ and $p_j = \frac{1}{t}(j = 1, \ldots, t)$. Then*

$$\underline{\Pi}_X(U) = \underline{\mathrm{H}}_X(U) = \frac{ND(1-\delta_2)^3}{3t|U|}, \tag{11}$$

where $D = |\bar{U}|$.

Corollary 2 *Let the restrictions of the theorem 8 are true. Then*

$$\underline{\Pi}_X(U) = \underline{\mathrm{H}}_X(U) = \frac{2N}{3t}(1-\delta_2)\underline{\xi}(s_t). \tag{12}$$

Theorem 9 *Let $X = \{a_1, \ldots, a_t\}$, $s_t \in G^{\delta}(\bar{L})$, $N(u) = N = Const$ and $p_j = \frac{1}{t}$ $(j = 1, \ldots, t)$. Then*

$$\overline{\Pi}_X(U) = \overline{\mathrm{H}}_X(U) = \frac{ND(1-\delta_2)^3}{3t|U|} + \frac{2ND\delta_2}{t|U|}, \tag{13}$$

where $D = |\bar{U}|$.

Corollary 3 *Let the restrictions of the theorem 9 are true. Then*

$$\overline{\Pi}_X(U) = \overline{\mathrm{H}}_X(U) = \frac{2N}{t(1+2\delta)}\left[\frac{(1-\delta_2)^3}{3} + 2\delta_2\right]\overline{\xi}(s_t). \tag{14}$$

By comparing the results of the theorem 6 and theorems 8 and 9 or the corollary 1 and corollaries 2 and 3, we see that for small significances δ, the main laws of our model of information retrieval are preserved. Therefore, we can use our technique of estimation of the degree of uncertainty and our model of information retrieval in fuzzy (linguistic) data bases in practical tasks, since we have shown it to be stable.

Based on these results we can propose the following method for optimal information granulation for information retrieval.

4.4 *Method for Optimal Information Granulation for Information Retrieval*

1. All the "reasonable" sets of linguistic values are formulated;
2. Each of such sets is represented in the form of $G(L)$;
3. For each set the measure of uncertainty $\xi(s_t)$ is calculated;
4. As the optimum set use of which would provide the maximum indices of quality of information retrieval we select the one which give us

$$\min_{s_t \in G(L)} \frac{\xi(s_t)}{t}$$

Stability of the measure of the losses of information ($\Pi_X(U)$) and of information noises ($H_X(U)$) (Theorems 8 and 9) allows us to use this method in practical applications.

5 Optimal Task-Driven Information Granulation

As we can see from Sects. 3.4 and 4.4, optimal collections of granules for problems 1 and 2 are different. We cannot propose a "universal" optimal fuzzy information granulation procedure for designers of Big Data systems. But we can propose an optimal such a procedure for some certain task.

How we can do this? At first, we need to formalize quality of the solution (see Sects. 3.2 and 4.2). For example, if we analyze fuzzy pattern recognition system, we can define quality of the solution by the following way.

Let we have K classes $C_1, \ldots, C_K$. Result of algorithm A for granulation $s_t \in G(L)$ and object $u*$ is membership function $\mu_{A,s_t}(u^*) = \left(\mu_{c_1}(u^*), \ldots, \mu_{c_K}(u^*)\right)$. The best result (no any uncertainty) is situation when $\exists i(1 \le i \le K) : \mu_{c_i}$ $(u^*) = 1, \mu_{c_j}(u^*) = 0 \forall j \neq i(1 \le j \le K)$. The worst result (maximal uncertainty) is situation when $\forall i(1 \le i \le K)\mu_{c_i}(u^*) = 0.5$. The simplest degree of uncertainty $\mu_{A,s_t}(u^*)$ is $\xi\left(\mu_{A,s_t}(u^*)\right) = 1 - |2\mu_{A,s_t}(u^*) - 1|$ (FLS defined by one membership

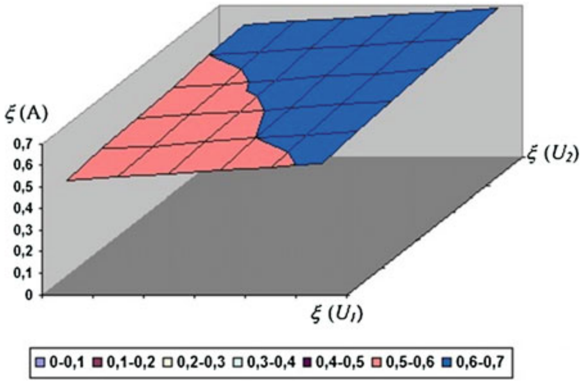


Fig. 7 Approximation of $\xi\left(\mu_{A,s_t}\right)$

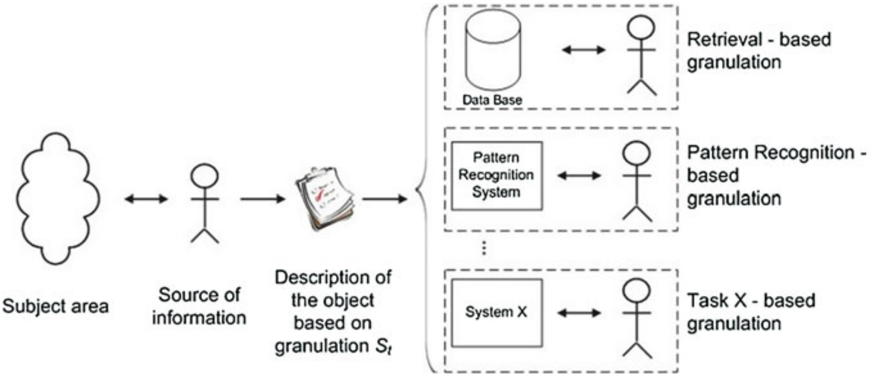

Data Base
Retrieval - based granulation
Pattern Recognition System
Pattern Recognition - based granulation
System X
Task X - based granulation
Subject area
Source of information
Description of the object based on granulation S_t

Fig. 8 Task-driven granulation

function—[22]). By analogy with (4) we can define the quality of pattern recognition by algorithm A for granulation s_t as $\xi\left(\mu_{A,s_t}\right) = \int_U p(u)\left(1 - \left|2\mu_{A,s_t}(u) - 1\right|\right)du$, where $p(u)$ is frequency function of u.

For certain algorithm A we can find the best granulation s_t or, opposite, the best algorithm for fixed granulation. Fuzzy rule-based pattern recognition algorithms are investigated in [25]. We have found dependences like theorem 7 for simplest cases only. In this situation we can use computational modelling techniques for approximation $\xi\left(\mu_{A,s_t}\right)$ [25]. One of the results of such modelling for objects with two attributes is presented in Fig. 7.

Based on the results above, we can provide recommendations how we can use similar approach for different type of tasks (information retrieval, pattern recognition, data mining [26], etc.—see Fig. 8).

1. Formalize of the quality of the task's solution.
2. Try to find dependence of the quality functional and granulation or use computational techniques for approximation the quality functional.
3. Choose granulation which provide maximum of the quality functional.

6 Concluding Remarks

We believe that big part of Big Data is a human-created data. For this type of data we need fuzzy information granulation. We can do this granulation be different ways. Which collection of granules is the best?

We have shown than for different tasks we can have a different optimal collection of granules.

For some tasks we can find formula which describes the dependence of the quality of the task's solution and granulation. In this case we can calculate the optimal granulation. For some tasks we cannot find such a formula. In this case we can use computational modelling for approximation of the quality functional.

If we cannot formalize the quality of the task's solution, or use Big Data system for different classes of tasks, my recommendation is to use optimal granulation for human-created data (Sect. 3.4). In this case we can guarantee that the number of situations when one real object has more than one image in our Big Data model (for example, in social networks), or different real objects have the same image, will be minimal. Accordingly, we will have a maximal possible adequacy of the data as a model of real world. Stability of the measure of uncertainty (Theorem 5) allows us to use this method in practical applications.

I do hope that this approach will be useful for designers of Big Data and help them to develop systems with highest quality of the solutions for different tasks.

Taking an opportunity, I would like to express my thanks to Professor Witold Pedrycz and Professor Shy-Ming Chen for the great idea to prepare this book and valuable remarks.

References

1. Big data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute. http://www.mckinsey.com/insights/mgi/research/technology_and_innovation/big_data_the_next_frontier_for_innovation. May 2011
2. EMC Big Data. http://www.emc.com/big-data/index.htm
3. IBM.: Bringing big data to the enterprise. http://www-01.ibm.com/software/data/bigdata/
4. Oracle and Big Data.: Transform your business with big data. http://www.oracle.com/us/technologies/big-data/index.html
5. Big Data, Big Impact: New possibilities for international development. World Economic Forum, Davos, Switzerland. http://www3.weforum.org/docs/WEF_TC_MFS_BigDataBigImpact_Briefing_2012.pdf. Jan 2012

6. Beyer, M.: Gartner says solving ‘big data’ challenge involves more than just managing volumes of data. http://www.gartner.com/newsroom/id/1731916. 27 June 2011
7. Nature 455, 1 (4 Sept 2008). doi:10.1038/455001a. Published online 3 Sept 2008
8. Microsoft Big Data.: http://www.microsoft.com/sqlserver/en/us/solutions-technologies/business-intelligence/big-data.aspx
9. HP Big Data Solutions.: http://www8.hp.com/us/en/business-solutions/big-data.html
10. Boyd E.B.: The challenges of moving to a big-data mindset. http://www.proformative.com/articles/challenges-moving-big-data-mindset. 30 Apr 2013
11. Lohr S.: The age of big data. New Your Times. http://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html?pagewanted=all&_r=0. 11 Feb 2012
12. Satell Greg.: Companies that can’t figure out data are getting left behind. Business Insider. http://www.businessinsider.com/how-big-data-affects-strategy-2013-8. 25 Aug 2013
13. Morris, J.: Top 10 categories for big data sources and mining technologies. http://www.zdnet.com/top-10-categories-for-big-data-sources-and-mining-technologies-7000000926/
14. Chuvakin, A.: Big data analytics mindset—what is it? Gartner Blog Network. http://blogs.gartner.com/anton-chuvakin/2013/11/18/big-data-analytics-mindset-what-is-it. 18 Nov 2013
15. Hollis, C.: Understanding the big data analytics mindset. Chuck’s blog. http://chucksblog.emc.com/chucks_blog/2011/12/understanding-the-big-data-analytics-mindset.html. 20 Dec 2011
16. Boyd, D., Crawford, K.: Six provocations for big data. A decade in internet time: symposium on the dynamics of the internet and society. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1926431. Sept 2011
17. Zadeh, L.A.: Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets Syst. **90**, 111–127 (1997)
18. Pedrycz, W.: Granular Computing: Analysis and Design of Intelligent Systems, p. 287. CRC Press/Francis Taylor, Boca Raton (2013)
19. Community cleverness required: Editorial. Nature **455**, 1. http://www.nature.com/nature/journal/v455/n7209/pdf/455001a.pdf. 4 Sept 2008
20. Ryjov, A.: The degree of fuzziness of fuzzy descriptions. In: Krushinsky, L.V., Yablonsky, S.V., Lupanov, O.B. (eds.) Mathematical Cybernetics and Its Application to Biology, pp. 60–77. Moscow University Publishing, Moscow (1987)
21. Ryjov, A.: Fuzzy linguistic scales: definition. Properties and applications. In: Reznik, L., Kreinovich, V. (eds.) Soft computing in measurement and information acquisition, pp. 23–38. Springer, Berlin (2003)
22. Ryjov, A.: The principles of fuzzy set theory and measurement of fuzziness. Dialog-MSU, Moscow, 116 p. (1998)
23. Ryjov, A.: Modeling and optimization of information retrieval for perception-based information. Brain Informatics. In: Zanzotto, F., Tsumoto, S., Taatgen, N., Yao, Y.Y. (eds.) International conference, BI 2012, proceedings, Dec 2012. doi:http://link.springer.com/chapter/10.1007/978-3-642-35139-6_14
24. Ryjov, A.: Models of information retrieval in fuzzy environment. Publishing house of Center of applied research, department of mechanics and mathematics, MSU, Moscow, 96 p. (2004)
25. Ryjov, A.: Quality of classification for fuzzy rule-based classifier. Intell. Syst. **9**, 253–264 (2005)
26. Rastorguev, V., Ryjov, A. Fuzzy associative rules in information monitoring systems. International Conference, Intelligent Systems 2006, Proceedings. (September, 2006)

Unified Framework for Construction of Rule Based Classification Systems

Han Liu, Alexander Gegov and Frederic Stahl

Abstract Automatic generation of classification rules has been an increasingly popular technique in commercial applications such as Big Data analytics, rule based expert systems and decision making systems. However, a principal problem that arises with most methods for generation of classification rules is the overfitting of training data. When Big Data is dealt with, this may result in the generation of a large number of complex rules. This may not only increase computational cost but also lower the accuracy in predicting further unseen instances. This has led to the necessity of developing pruning methods for the simplification of rules. In addition, classification rules are used further to make predictions after the completion of their generation. As efficiency is concerned, it is expected to find the first rule that fires as soon as possible by searching through a rule set. Thus a suitable structure is required to represent the rule set effectively. In this chapter, the authors introduce a unified framework for construction of rule based classification systems consisting of three operations on Big Data: rule generation, rule simplification and rule representation. The authors also review some existing methods and techniques used for each of the three operations and highlight their limitations. They introduce some novel methods and techniques developed by them recently. These methods and techniques are also discussed in comparison to existing ones with respect to efficient processing of Big Data.

H. Liu (✉) · A. Gegov
School of Computing, University of Portsmouth, Buckingham Building,
Lion Terrace, Portsmouth PO1 3HE, UK
e-mail: han.liu@port.ac.uk

A. Gegov
e-mail: alexander.gegov@port.ac.uk

F. Stahl
School of Systems Engineering, University of Reading, White Knights, 225
Reading RG6 6AY, UK
e-mail: f.t.stahl@reading.ac.uk

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_10,

Keywords Data mining · Machine learning · Rule based systems · Rule based classification · Information granularity · Big data · Computational intelligence

1 Introduction

Automatic induction of classification rules has been increasingly popular in commercial applications such as Big Data analytics, rule based expert systems and predictive decision making systems. The methods of classification rule generation can be divided into two categories: ‘divide and conquer’ and ‘separate and conquer’. The former is also known as Top-Down Induction of Decision Trees (TDIDT) [1], which generates classification rules in the intermediate form of a decision tree such as ID3 [1], C4.5 and C5.0. The latter is also known as covering approach [2], which generates if-then rules directly from training instances such as Prism [3]. A series of experiments have shown that Prism achieves a similar level of accuracy compared with TDIDT and can even outperform TDIDT in some cases [4].

However, a principal problem [5] that arises with most methods for generation of classification rules is the overfitting of training data. When the training data is large, this may result in the generation of a large number of complex rules. This may not only increase computational cost but also lower the accuracy in predicting further unseen instances. This has motivated the development of pruning algorithms with respect to the reduction of overfitting. Pruning methods can be subdivided into two categories—pre-pruning [6], which truncates rules during rule generation, and post-pruning [6], which generates a whole set of rules and then remove a number of rules and rule terms, by using statistical or other tests [5]. A family of pruning algorithms are based on J-measure [7] used as information theoretic means of quantifying the theoretical information content of a rule. This is based on a working hypothesis [8] that rules with high information content (value of J-measure) are likely to have a high level of predictive accuracy. Two existing J-measure based pruning algorithms are J-pruning [5] and Jmax-pruning [9, 10], which have been successfully applied to Prism for the reduction of overfitting.

The main objective in prediction stage is to find the first rule that fires by searching through a rule set. As efficiency is concerned, a suitable structure is required to effectively represent a rule set that is generated by learning from Big Data. The existing rule representations include tree and list. Tree representation is mainly used to represent rule sets generated by ‘divide and conquer’ approach in the form of decision trees. It has root and internal nodes representing attributes and leaf nodes representing classifications as well as branches representing attribute values. On the other hand, list representation is commonly used to represent rules generated by ‘separate and conquer’ approach in the form of ‘if-then’ rules.

As the relevance of above operations, the authors have recently developed a unified framework consisting of these operations for the construction of rule based classification systems. On the other hand, it is stated in [11] that “Big Data is a popular term used to describe the exponential growth and availability of data, both structured and unstructured. And Big Data may be as important to business—and society—as the Internet has become." This is due to the following reasons [11]:

- More data may lead to more accurate analyses.
- More accurate analyses may lead to more confident decision making.
- Better decisions can mean greater operational efficiencies, cost reductions and reduced risk.

IBM defines that Big Data is characterised by four Vs [12]:

- **Volume**—terabytes, petabytes, or more
- **Velocity**—data in motion or streaming data
- **Variety**—structured and unstructured data of all types—text, sensor data, audio, video, click streams, log files and more
- **Veracity**—the degree to which data can be trusted

Therefore, this chapter aims to introduce a framework for construction of rule based classification systems particularly on Big Data and to review some existing methods and techniques involved in each of the three operations namely generation, simplification and representation highlighting their limitations. The chapter also introduces some novel methods and techniques that are based on information theory and that may overcome the limitations of those methods reviewed with respect to effective and efficient processing of Big Data.

The rest of the chapter is organized as follows. Section 2 reviews Prism algorithm and identifies its limitations with respect to rule generation. It also discusses in what way J-pruning and Jmax-pruning help Prism and other rule based classifiers overcome overfitting with respect to rule simplification and the efficiency that list and tree representation achieve at prediction stage. Section 3 introduces three novel methods and techniques developed by the authors in their more recent research. It includes Information Entropy Based Rule Generation (IEBRG), Jmid-pruning and networked rule representation. The methods and techniques are discussed further in comparison to existing ones with respect to effective and efficient processing of Big Data in Sect. 4. Section 5 summarises the completed work reflecting the potential use to real world problems and highlights further directions.

2 Related Work

As mentioned in Sect. 1, a unified framework for the construction of rule based classification systems involves three operations: rule generation, rule simplification and rule representation. In this section, Prism is selected as a representative for the methods of classification rules generation with the reason that Prism is more noise-tolerant and achieves a higher predictive accuracy comparing to decision trees in some special cases but also can perform similar accuracy to decision trees in most cases. Furthermore, J-pruning and Jmax-pruning are reviewed because only the two existing pruning methods have been applied to Prism for rule simplification. In addition, rules that are generated by 'divide and conquer' approach are automatically represented in the form of decision trees and rules generated by 'separate and conquer' approach are directly represented by a linear list in the form of 'if-then' rules. However, they both have their limitations as criticised in [3] and identified by the authors respectively. These methods and techniques are described in the following subsections highlighting the limitations of them to show the motivation for the development of those novel methods and techniques which are further presented in Sect. 3.

2.1 Prism Method

The Prism method was introduced by Cendrowska in [3] and the basic procedure of the underlying Prism algorithm is illustrated in Fig. 1. This algorithm is primarily aimed at avoiding the generation of complex rules with many redundant terms [5] such as the 'replicated subtree problem' [3] that arises with decision trees as illustrated in Fig. 2.

The original Prism algorithm cannot directly handle continuous attributes as it is based on the assumption that all attributes in a training set are categorical. When continuous attributes are actually present in a dataset, these attributes should be discretized by preprocessing the dataset prior to generating classification rules [6, 9, 10]. In addition, Bramer's Inducer Software handles continuous attributes as described in [6, 9, 10] and in Sect. 3.

On the other hand, the original Prism algorithm does not take clashes into account, i.e. a set of instances in a subset of a training set that are identical apart from being assigned to different classes but cannot be separated further [6, 10]. However, the Inducer Software implementation [13] of Prism can handle clashes and the strategy of handling a clash is illustrated in Fig. 3.

Another problem that arises with Prism is tie-breaking, i.e. if there are two or more attribute-value pairs which have equal highest probability in a subset (see step 3 in Fig. 1). The original Prism algorithm makes an arbitrary choice in step 4 as illustrated in Fig. 1 whereas the Inducer Software makes the choice using the highest total target class frequency [6].

In addition, Bramer pointed out that the original Prism algorithm always deletes instances covered by those rules generated so far and then restores the training set to its original size after the completion of rule generation for class *i* and before the start for class $i + 1$. This undoubtedly increases the number of iterations resulting in high computational cost [14] when the training data is very large. For the purpose of increasing the computational efficiency, a modified version of Prism, called PrismTCS, was developed by Bramer [15]. PrismTCS always chooses the minority class as the target class pre-assigned to a rule being generated as its consequence. Besides this, it does not reset the dataset to its original state and

Execute the following steps for each classification (*class*= *i*) in turn and on the original training data *S*:

1. *S'*=*S*.

2. Remove all instances from *S'* that are covered from the rules induced so far. If *S'* is empty then stop inducing further rules

3. Calculate the conditional probability from *S'* for *class*=*i* for each *attribute-value pair*.

4. Select the *attribute-value pair* that covers *class*= *i* with the highest probability and remove all instances from *S'* that comprise the selected *attribute-value pair*

5. Repeat 3 and 4 until a subset is reached that only covers instances of *class*= *i* in S'. The induced rule is then the conjunction of all the *attribute-value pairs* selected.

Repeat 1-5 until all instances of *class i* have been removed

*For each rule, no one attribute can be selected twice during rule generation

Fig. 1 Basic prism algorithm [6]

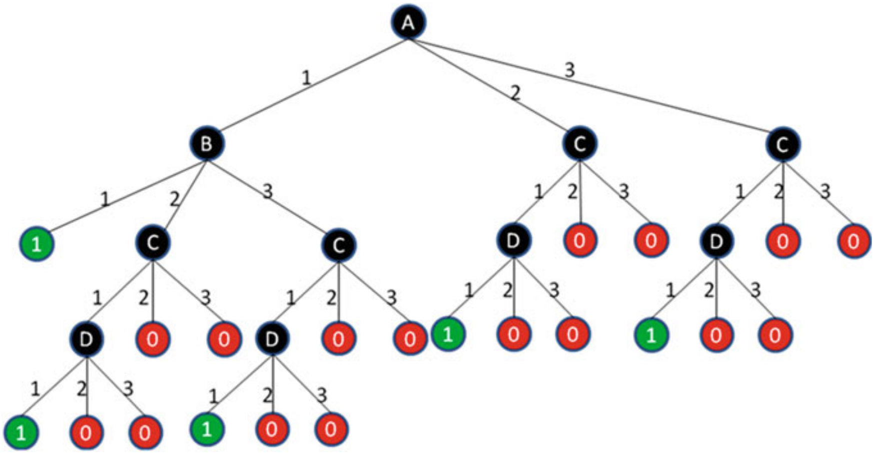


Fig. 2 Cendrowska's replicated subtree example [9, 10, 19]

If a clash occurs while generating rules for class *i*:

1. Determine the majority class for the subset of instances in the clash set.

2. If this majority class is target *class i*, then compute the induced rule by assigning all instances in the clash set to target class *i*. If it is not, discard the whole rule.

3. If the induced rule is discarded, then all instances that match the target class should be deleted from the training set before the start of the next rule induction.

If the rule is kept, then all instances in the clash set should be deleted from the training data.

Fig. 3 Dealing with clashes in Prism

introduces an order to each rule according to its importance [9, 10, 14]. Therefore, PrismTCS is not only faster in generating rules compared with the original Prism, but also provides a similar level of classification accuracy [9, 10, 15].

Prism algorithm also has some disadvantages. One of them is that the original version of Prism may generate a rule set which may result in a classification confliction in predicting unseen instances. This can be illustrated by the example below:

Rule 1: If x = 1 and y = 1 then class = a
Rule 2: If z = 1 then class = b

What should the classification be for an instance with x = 1, y = 1 and z = 1? One rule gives *class a*, the other one gives *class b*. We need a method to choose only one classification to classify the unseen instance [6]. Such a method is known as a conflict resolution strategy. Bramer mentioned in [6] that Prism uses the 'take the first rule that fires' strategy in dealing with the conflict problem and therefore it is required to generate the most important rules first. However, the original Prism cannot actually introduce an order to a rule according to its importance as each of those rules with a different target class is independent from each other. As mentioned above, this version of Prism would restore the training set to its original size after the completion of rule generation for class *i* and before the start for class *i + 1*. This indicates the rule generation for each class may be done in parallel so the algorithm cannot directly rank the importance among rules. Thus the 'take the first rule that fires' strategy may not deal with the classification confliction well. The PrismTCS does not restore dataset to its original state unlike original Prism and thus can introduce the order to a rule for its importance. This problem is partially resolved but PrismTCS may potentially lead to underfitting of a rule set. PrismTCS always chooses the minority class in the current training set as the target class of the rule being generated. Since the training set is never restored to its original size as mentioned above, it can be proven that one class could always be selected as target class until all instances of this class have been deleted from the training set because the instances of this minority class covered by the current rule generated should be removed prior to generating the next rule. This case may result in that the majority class in the training set may not be necessarily selected as target class to generate a list of rules until the termination of the whole generation process. In this case, there is not even a single rule having the majority class as its consequence (right hand side of this rule). In some implementations, this problem has been partially solved by assigning a default class (usually majority class) in predicting unseen instances when there is not a single rule that can cover this instance. However, this should be based on the assumption that the training set is complete. Otherwise, the rule set may still underfit on training set as the conditions of classifying instances to the other classes are probably not strong enough. On the other hand, if a clash occurs, both the original Prism and PrismTCS would prefer to discard the whole rule rather than to assign the majority class, which is higher in importance, to the rule. As mentioned above, Prism may generally generate more general and less rules than a decision tree. One reason is potentially due to discarding rules. In addition, the clash may happen in two principal ways as follows:

1. One of the instances has at least one incorrect record for its attribute values or its classification [5].
2. The clash set has both (or all) instances correctly recorded but it is impossible to discriminate between (or among) them on the basis of the attributes recorded and thus it may be required to examine further values of attributes [6].

When there is noise present in datasets, Prism may be more robust than decision trees as mentioned above. However, if the reason that a clash occurs is not due to noise and the training set covers a large amount of data, then it may result in serious underfitting of the rule set by discarding rules as it will leave many unseen instances unclassified at prediction stage. The fact that Prism would decide to discard the rules in some cases is probably because it uses the so-called 'from effect to cause' approach. As mentioned above, each rule being generated should be pre-assigned a target class and then the conditions should be searched by adding terms (antecedents) until the adequacy conditions are met. Sometimes, it may not necessarily receive adequacy conditions even after all attributes have been examined. This indicates the current rule covers a clash set that contains instances of more than one class. If the target class is not the majority class, this indicates the search of causes is not successful so the algorithm decides to give up by discarding the incomplete rule and deleting all those instances that match the target class in order to avoid the same case to happen all over again [9, 10]. This actually not only increases the irrelevant computation cost but also results in underfitting of the rule set.

These limitations have motivated the development of a new method for the generation of classification rules which is further introduced in Sect. 3.1.

2.2 J-Pruning and Jmax-Pruning

As mentioned in Sect. 1, both J-pruning and Jmax-pruning are based on J-measure which was introduced by Smyth and Goodman [7] who justified the use of the J-measure as an information theoretic means of quantifying the theoretical information content of a rule.

According to the notation of [7], given a rule of the form *IF* $Y = y$ *THEN* $X = x$ can be measured in bits and is denoted by *J(X,* $Y = y$*)*.

$$J(X; Y = y) \ = p(y) \ \cdot j(X; Y = y) \tag{1}$$

$J(X;\ Y = y)$ is essentially a product of two terms as follows:

- $p(y)$, the probability that the left hand side of the rule (hypothesis) will occur.

- $j(X; Y = y)$, which is called the j-measure (with a lower case j) and measures the goodness-of-fit of a rule.

The j-measure, also known as the *cross-entropy*, is defined as:

$$\begin{aligned} j(X; Y = y) &= p(x|y) \cdot \log_2(p(x|y)/p(x)) + (1 - p(x|y)) \\ &\quad \cdot \log_2((1 - p(x|y))/(1 - p(x))) \end{aligned} \tag{2}$$

The value of *cross-entropy* depends upon two values [6]:

- $p(x)$: the probability that the consequence (right hand side) of the rule will be matched if there is no other information given. This is known as a priori probability of the rule consequence.
- $p(x|y)$: the probability that the consequence of the rule is matched if the given antecedents are satisfied. This is also read as *a posterior* probability of x given y.

Bramer mentioned in [5, 6] that the J-measure has two very helpful properties related to upper bounds as follows:

- It can be shown that $J(X; Y = y) \leq p(y) \cdot \log_2 (1/p(y))$. The maximum point of this expression can be found at $p(y) = 1/\mathrm{e}$. This can derive a maximum value, is ($\log_2$ (e)·(1/e)), i.e. approximately 0.5307 bits.
- More importantly, it can be proven that the value of the J-measure is never higher than the upper bound value illustrated in Eq. (3) whenever a rule is specialised by adding further terms to its left hand side.

$$J\mathrm{max} = p(y) \cdot \max\{p(x|y) \cdot \log_2(1/p(x)), (1 - p(x|y)) \cdot \log_2(1/1 - p(x))\} \tag{3}$$

Thus, there are no theoretical benefits to be gained by adding further terms to a rule when the value of the J-measure of this rule is equal to its corresponding Jmax-value. The application of Jmax is illustrated in Sect. 3.2.

When a rule is being generated, the J-value (value of J-measure) may increase or decrease after specialising the rule by adding a new term. Both pruning algorithms (J-pruning and Jmax-pruning) expect to find the global maximum of J-value for the rule. Each rule has a complexity degree which is the number of terms. The increase of complexity degree may lead the J-value of this rule to increase or decrease. The aim of pruning algorithms is to find the complexity degree corresponding to the global maximum of J-value as illustrated in Fig. 4 using a fictitious example.

However, the two pruning algorithms mentioned above search the global maximum of J-value with different strategies:

- J-pruning: monitor the change pattern of J-value and stop rule generation once it goes down. i.e. it will stop rule generation when complexity degree is X_1 as illustrated in Fig. 4 because the J-value is going to decrease afterwards. The final rule generated is with the complexity degree X_1 (having the first X_1 rule terms).

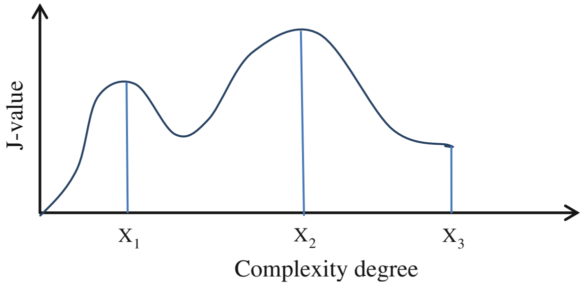


Fig. 4 Relationship between complexity degree and J-value (case 1)

- Jmax-pruning: monitor and record the highest J-value observed so far until the completion of rule's generation. i.e. it will stop rule generation when the complexity is X_3 as illustrated in Fig. 4 and reduce the complexity degree subsequently until the degree is X_2 by removing those rule terms afterwards. The final rule is with the complexity degree X_2.

J-pruning is a pre-pruning method because the pruning action is taken during rule generation. It was developed by Bramer [5] and its basic idea is illustrated in Algorithm 1.

Algorithm 1 J-pruning for Prism algorithms

```
Rule r = new Rule;
Boolean rule_Incomplete = true;
Do While (rule_Incomplete){
    Term t = generate new term;
    compute J_value of r if appending t;
    IF(r.current_J_value > J_value){
        do not append t to r;
        invoke clash handling for r;
        rule_Incomplete = false;
    }ELSE{
        r.current_J_value = J_value;
        append t to r;
    }
}
```

J-pruning achieves relatively good results as indicated in [5]. However, Stahl and Bramer pointed out in [9, 10] that J-pruning does not exploit the J-measure to its full potential. This is because this method immediately stops the generation process as soon as the J-measure goes down after a new term is added to the rule as illustrated in Fig. 4. In fact, it is theoretically possible that the J-measure may go down and go up again after further terms are added to the rule. This indicates the pruning action may be taken too early. The fact that J-pruning may achieve relatively good results could be explained by the assumption that it does not happen very often that the J-value goes down and then goes up again. A possible case is that there is only one local maximum of J-value as illustrated in Fig. 5. It also

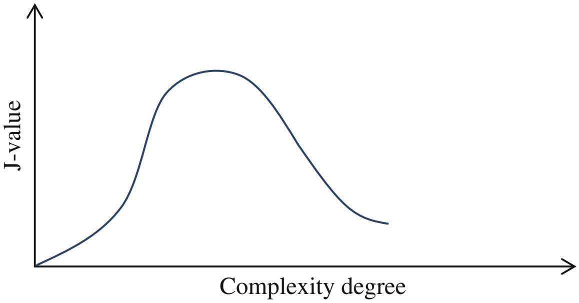


Fig. 5 Relationship between complexity degree and J-value (case 2)

indicates that J-pruning may even result in underfitting due to over-generalised rules. This is because the pruning action may be taken too early resulting in too general rules being generated. This motivated the development of a new pruning method, called Jmax-pruning, which was proposed by one of the authors of this chapter [9, 10], in order to exploit the J-measure to its full potential.

Jmax-pruning can be seen as a hybrid between pre-pruning and post-pruning. With regard to each generated rule, each individual rule is actually post-pruned after the completion of the generation for that rule. However, with respect to the whole classifier (whole rule set) it is a pre-pruning approach as there is no further pruning required after all rules have been induced.

The basic idea of Jmax-pruning is illustrated in Algorithm 2.

Algorithm 2 Jmax-pruning for Prism algorithms

```
 Rule r = new Rule;
 Boolean rule_Incomplete = true;
 term_index = 0;
 Do While (rule_Incomplete){
 Term t = generate new term;
 term_index++;
 append t to r;
 compute J_value of r;
 IF(J_value > best_J_Value){
    best_J_Value = J_Value;
   best_term_index = term_index;
 }
 IF(No more rule terms can be induced){
    cut r back to rule best_term_index;
    invoke clash handling for r;
    rule_Incomplete = false;
 }
}
```

A series of experiments have shown that Jmax-pruning outperforms J-pruning in some cases [9, 10] when there are more than one local maximum and the first one is not the global maximum as illustrated in Fig. 4. However, it performs the same

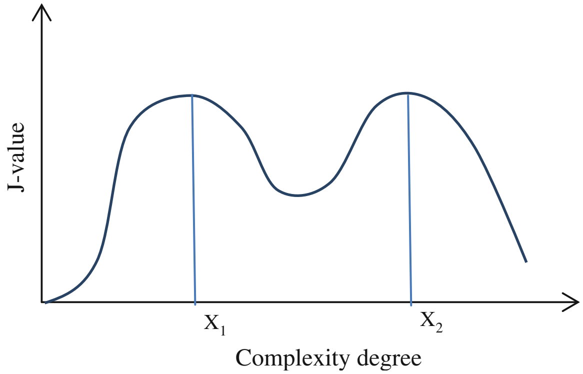


Fig. 6 Relationship between complexity degree and J-value (case 3)

as J-pruning in other cases [9, 10] when there is only one local maximum as illustrated in Fig. 5 or the first one of local maxima is also the global maximum.

However, Jmax-pruning may be computationally relatively expensive as each rule generated by this method is post-pruned. The pruning action could be taken earlier during the rule generation and thus speed up the rule generation when Big Data is used for training. This could be achieved by making use of the Jmax value as introduced above.

On the other hand, a special case may need to be taken into account when Prism is used as the classifier. This case is referred to as tie-breaking which is if there is more than one global maximum for the J-value during rule generation as illustrated in Fig. 6.

As mentioned in Sect. 2.1, Prism prefers to discard a rule rather than assign it to a majority class when a clash occurs. Therefore, it may even lead to underfitting of the induced rule set if a pruning method attempts to reduce the overfitting by pruning rules but unfortunately results in discarding rules. If this case is taken into account, it is worth to determine properly which one of the global maximum points to be chosen as the start point of pruning in order to avoid over-discarding rules. In other words, according to Fig. 6, it needs to determine to choose either X_1 or X_2 as the start point for removing all rule terms afterwards.

With regards to this issue, Jmax-pruning always chooses to take X_1 (the first global maximum point) as the start point of pruning and to remove all rule terms generated afterwards. It may potentially lead to underfitting as it is possible that the rule is being discarded after handling a clash if X_1 is chosen but is being kept otherwise. In addition, another type of tie-breaking may arise with the case as illustrated below:

Let the current rule's last added rule term be denoted t_i, and the previously added rule term be denoted t_{i-1}. Then a tie break happens if J-value at t_i is less than that at t_{i-1} and Jmax-value at t_i equals J-value at t_{i-1}. It is also illustrated by an example (**Rule 1**) below.

Rule 1: If x = 1 and y = 1 and z = 1 then class = 1;
After adding first term:
If x = 1 then class = 1; (J = 0.33, Jmax = 0.55)
After adding second term:
If x = 1 and y = 1 then class = 1; (J = 0.21; Jmax = 0.33)

However, the two cases about tie-breaking mentioned above are not very likely to happen. As the basis of above descriptions about limitations of J-pruning and Jmax-pruning, it has motivated the development of a new pruning algorithm to overcome the limitations of J-pruning and Jmax-pruning with respects to underfitting and computational efficiency. The new pruning algorithm is further introduced in Sect. 3.2.

2.3 Decision Tree and Linear List Representation

As mentioned in Sect. 1, decision tree is an automatic representation for classification rules generated by 'divide and conquer' approach. However, the representation has been criticized by Cendrowska and identified as a major cause of overfitting in [3] as illustrated in Fig. 2. It was also pointed in [16] that it is required to examine the whole tree in order to extract rules about a single classification in the worst case. This drawback on representation has made it difficult to manipulate for expert systems. It has thus motivated the direct use of 'if then' rules represented by a linear list structure. However, simulation in this representation is run in linear search with the time complexity O (n) while the total number of rule terms is used as the input size (n). This is because list representation works in linear search by going through rule by rule in an outer loop; and by going through term by term for each rule in an inner loop. It implies it may have to go through the whole rule set to find the first rule that fires in the worst case. This may lead to huge computational costs when the representation is used to represent a rule set generated by learning from Big Data.

As the basis of above description about limitations of tree and list representation, it has motivated the development of a new representation of classification rules which performs a level of efficiency higher than linear time in time complexity. This new representation is further described in Sect. 3.3.

3 Novel Methods and Techniques

Section 2 has reviewed a representative rule generation method called Prism, two J-measure based pruning algorithms namely J-pruning and Jmax-pruning and two types of representation of classification rules namely tree and list. It has also highlighted their limitations so this section explores a novel rule generation

1. Calculate the conditional entropy of each attribute-value pair in the current subset
2. Select the attribute-value pair with the smallest entropy to spilt on, i.e. remove all other instances that do not comprise the attribute-value pair.
3. Repeat step 1 and 2 until the current subset contains only instances of one class (the entropy of the resulting subset is zero).
4. Remove all instances covered by this rule.

Repeat 1-4 until there are no instances remaining in the training set.

* For each rule, no one attribute can be selected more than once during generation.

Fig. 7 IEBRG algorithm

method called Information Entropy Based Rule Generation (IEBRG); a novel J-measure based pruning algorithm called Jmid-pruning and a novel representation of classification rules called Rule Based Classification Networks.

3.1 Information Entropy Based Rule Generation

Information Entropy Based Rule Generation is a method of classification rules generation following ‘separate and conquer’ approach and has been recently developed in [17]. This method tends to avoid underfitting and redundant computational efforts.

3.1.1 Essence

This method is attribute-value-oriented like Prism but it uses the ‘from cause to effect’ approach. In other words, it does not have a target class pre-assigned to the rule being generated. The main difference with respect to Prism is that IEBRG focuses mainly on minimising the uncertainty for each rule being generated no matter what the target class is. A popular technique used to measure the uncertainty is information entropy introduced by Shannon in [18]. The basic idea of IEBRG is illustrated in Fig. 7.

3.1.2 Justification

As mentioned in Sect. 2.1, all versions of Prism need to have a target class pre-assigned to the rule being generated. In addition, an attribute might be not relevant to each particular classification and sometimes only one value of an attribute is relevant [16]. Therefore, the Prism method chooses to pay more attention to the relationship between attribute-value pair and a particular class. However, the class

Table 1 Lens 24 dataset example

Class label	Tears = 1	Tears = 2
Class = 1	0	4
Class = 2	0	5
Class = 3	12	3
Total	12	12

to which the attribute-value pair is highly relevant is probably unknown, as can be seen from the example in Table 1 below with reference to the lens 24 dataset reconstructed by Bramer in [6]. This dataset shows that P (class = 3|tears = 1) = 1 illustrated by the frequency table for attribute “tears”. The best rule generated first would be “if tears = 1 then class = 3”.

This indicates that the attribute-value “tears = 1” is only relevant to class 3. However, this is actually not known before the rule generation. According to PrismTCS strategy, the first rule being generated would select “class = 1” as target class as it is the minority class (Frequency = 4). Original Prism may select class 1 as well because it is in a smaller index. As described in [6], the first rule generated by Original Prism is “if astig = 2 and tears = 2 and age = 1 then class = 1”. It indicates the computational efficiency is slightly worse than expected and the resulting rule is more complex. When Big Data is used for training, the Prism method may be even likely to generate an incomplete rule covering a clash set as mentioned in Sect. 2.1 if the target class assigned is not a good fit to some of those attribute-value pairs in the current training set. Then the whole rule may be discarded resulting in underfitting and redundant computational effort.

In order to find a better strategy for reducing the computational cost, the authors proposed the method in [17]. In this technique, the first iteration of the rule generation process for the “lens 24” dataset can make the resulting subset’s entropy reach 0. Thus the first rule generation is complete and its rule is represented by “if tears = 1 then class = 3”.

In comparison to the Prism family, this algorithm may reduce significantly the computational cost when Big Data is being dealt with. In addition, in contrast to Prism, the IEBRG method deals with clashes (introduced in Sect. 3.1.3) by assigning a majority class in the clash set to the current rule. This may potentially reduce the underfiting of rule set thus reducing the number of unclassified instances although it may increase the number of misclassified instances. On the other hand, the IEBRG may also have the potential to avoid occurring clashes better compared with Prism.

3.1.3 Dealing with Clashes

There are two principal ways of dealing with clashes mentioned in [6] as follows:

1. Majority voting: to assign the most common classification of the instances in the clash set to the current rule.

2. Discarding: to discard the whole rule currently being generated

In [17], the authors choose ‘majority voting’ as the strategy of dealing with this problem as the objective of [17] is mainly to validate this method and to find its potential in improving accuracy and computation efficiency as much as possible.

3.1.4 Dealing with Tie-Breaking on Conditional Entropy and Conflict

The tie-breaking problem on conditional entropy is solved by deciding which attribute-value pair is to be selected to split the current subset when there are two or more attribute-value pairs that equally well match the selection condition. In the IEBRG method, this problem may occur when two or more attribute-value pairs have the same smallest entropy value. The strategy is the same as the one applied to Prism by taking the one with the highest total frequency as introduced by Bramer [6].

The classification conflict problem may occur to modular classification rule generator such as Prism. Similarly, the IEBRG may also face this problem. The authors choose the ‘take the first rule that fires’ strategy which is already mentioned in Sect. 2.3 because this method may potentially generate the most important rules first. Consider the example below:

- Rule 1: if x = 1 and y = 1 then class = 1;
- Rule 2: if x = 1 then class = 2;

This seems as if there is a conflict problem but the two rules can be ordered as rule 1 is more important. In other words, the second rule can be represented in the following way:

Rule 2: if x = 1 and y $\neq$ 1 then class = 2;

This may indicate that after the first rule has been generated, all instances covered by the rule have been deleted from training set; then the two conditions ‘x = 1’ and ‘y = 1’ cannot be met simultaneously any more. Thus the first rule is more important than the second one.

3.2 Jmid-Pruning

The authors have recently mentioned in [19] that neither J-pruning nor Jmax-pruning exploit the J-measure to its full potential and they may lead to underfitting. In addition, Jmax-pruning is computationally relatively expensive. Therefore, the authors developed a novel pruning algorithm that avoids underfitting and unnecessary rule term inductions while at the same time rules are being pruned for reducing overfitting [19].

3.2.1 Essence

The Jmid-pruning is a modified version of the J-measure based pruning algorithm Jmax-pruning. It not only monitors and records the highest J-value observed so far but also measures the potentially highest J-value that may be achieved eventually by making use of the Jmax value highlighted in Sect. 2.2 in comparison to Jmax-pruning. The basic concept of this algorithm is illustrated in Algorithm 3.

Algorithm 3 Jmid-pruning for Prism algorithms

```
        Rule r = new Rule;
        Boolean rule_Incomplete = true;
        term_index = 0;
        Do While (rule_Incomplete){
        Term t = generate new term;
        term_index++;
        append t to r;
        compute J_value of r;
        IF(J_value > best_J_Value){
           best_J_Value = J_Value;
          best_J_term_index = term_index;

          record current_marjority_class;
    }
          compute Jmax_value of r;
     IF(best_J_value> Jmax_value){
           do not append t to r;
           cut r back to rule best_term_index;
       invoke clash handling for r;
       rule_Incomplete = false;
          } ELSE{
        append t to r;
      }
        IF(No more rule terms can be induced){
       cut r back to rule best_term_index;
       invoke clash handling for r;
       rule_Incomplete = false;
    }
}
```

3.2.2 Justification

The Jmid-pruning aims to avoid underfitting and unnecessary computational effort especially when Big Data is used for training. In fact, J-pruning and Jmax-pruning do not actually make use of Jmax value to measure the potential search space of gaining benefits.

Let us consider an example [8] using the lense24 dataset. There is a rule generated as follows:

If tears = 2 and astig = 1 and age = 3 and specRx = 1 then class = 3;

After adding the four terms subsequently, the corresponding J and Jmax values change in the trend as follows:

If tears = 2 then class = 3; (J = 0.210, Jmax = 0.531)

If tears = 2 and astig = 1 then class = 3; (J = 0.161, Jmax = 0.295)

If tears = 2 and astig = 1 and age = 3 then class = 3; (J = 0.004, Jmax = 0.059)

If tears = 2 and astig = 1 and age = 3 and specRx = 1 then class = 3; (J = 0.028, Jmax = 0.028)

In this example, all of the three algorithms would provide the same simplified rule that is: if tears = 2 then class = 3; this is because the highest J-value has been given after adding the first term (tears = 2). However, the computational efficiency would be different in the three methods. J-pruning would decide to stop the generation after the second term (astig = 1) is added as the J-value goes down after the second term (astig = 1) is added. In contrast, Jmax-pruning would stop when the rule is complete. In other words, the generation would be stopped after the fourth (last) term is added and then the terms (astig = 1, age = 3 and specRx = 1) will be removed. In addition, Jmid-pruning would decide to stop the generation after the third term is added as the value of Jmax (0.295) is still higher than the J-value (0.210) given after the first term (tears = 2) is added although its corresponding J-value (0.161) decreases; however, the generation should be stopped after the third term (age = 3) is added as both J (0.004) and Jmax (0.059) values are lower than the J-value (0.161) computed after the second term (astig = 1) is added although the J-value could still increase up to 0.059.

On the basis of the description above, J-pruning would be the most efficient and Jmid-pruning is more efficient than Jmax-pruning. However, it seems J-pruning may prune rules too early when the training data is large as mentioned in Sect. 2.2. For example, one of the rules [9, 10] generated from the Soybean dataset [20] is:

If temp = norm and same-lst-sev-yrs = whole-field and crop-hist = same-lst-two-yrs then class = frog-eye-leaf-spot;

First term:

If temp = norm then class = frog-eye-leaf-spot; (J = 0.00113, Jmax = 0.02315)

Second term:

If temp = norm and same-lst-sev-yrs = whole-field then class = frog-eye-leaf-spot; (J = 0.00032, Jmax = 0.01157)

Third term:

If temp = norm and same-lst-sev-yrs = whole-field and crop-hist = same-lst-two-yrs then class = frog-eye-leaf-spot; (J = 0.00578, Jmax = 0.00578)

In this case, both Jmax-pruning and Jmid-pruning would normally stop the generation when the rule is complete and take the complete rule: If temp = norm and same-lst-sev-yrs = whole-field and crop-hist = same-lst-two-yrs then class = frog-eye-leaf-spot; as the final rule with the highest J-value (0.00578). In contrast, J-pruning would stop the generation after the second term (same-lst-sev-yrs = whole-field) is added and take the rule: If temp = norm then class = frog-eye-leaf-spot; as the final rule with a lower J-value (0.00113 instead of 0.00578).

The other potential advantage of Jmid-pruning in comparison with Jmax-pruning is that Jmid-pruning may get more rules not being discarded later when tie-breaking on J-value happens as mentioned in Sect. 2.2. In this way, Jmid-pruning is better in avoiding underfitting of rule sets.

3.3 Rule Based Classification Networks

As mentioned in Sect. 2.3, both tree and list representations have their individual limitations. The authors have recently developed a networked representation of classification rules called rule based classification networks.

3.3.1 Essence

Let us see a set of rules based on Boolean logic below:

If x1 = 0 and x2 = 0 then class = 0;
If x1 = 0 and x2 = 1 then class = 0;
If x1 = 1 and x2 = 0 then class = 0;
If x1 = 1 and x2 = 1 then class = 1;

The corresponding networked representation is illustrated in Fig. 8. In this representation, x1 = 1 and x2 = 1 are supposed to be the two inputs respectively for simulation (prediction). Thus both 'x1' and 'x2' layers get green node labelled 1 and red node labelled 0 because each node in the layer x1 represents a value of attribute x1 and so does each node in layer x2. In addition, the two digits labelled to each of the connections between the nodes in layer x1 and x2 represent the index of rule and rule term respectively. In other words, the two digits '11' as illustrated below indicates it is for the first rule and the first term of the rule. It can be seen from the list of rules above that the first term of the first rule is 'x1 = 0'. However, the input value of x1 is 1 so the connection is coloured red as this condition is not met. In contrast, the connections labelled '31' and '41' respectively are both coloured green as the condition 'x1 = 1' is met. The same principle is also applied to the connections between the nodes in layer 'x2' and 'Rule Index'. As the two inputs are 'x1 = 1 and 'x2 = 1', the connections '31', '41' and '42' are coloured green and the node labelled 3 is green in the layer 'Rule Index' as well as the output is 1 in the layer 'Class'.

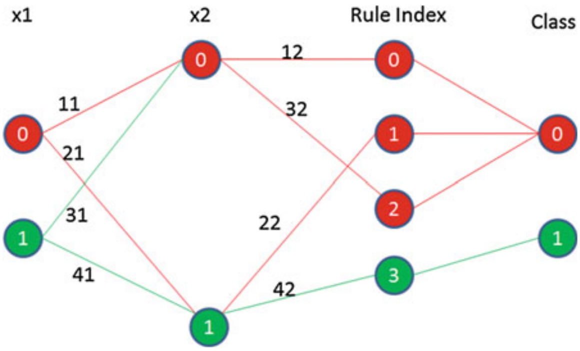


Fig. 8 Rule based classification networks

3.3.2 Justification

For Rule Based Classification Networks, simulation process is run by going through rule terms in divide and conquer search (i.e. only going through those terms that fire). The total number of terms is used as the input size of data (n) as same as used in linear list representation and thus the efficiency is O (log (n)). As can be seen from Fig. 8, it only takes three steps (going through connections '31', '41' and '42') to find the first rule that fires (the rule index is 3). This is because the input value of x1 is 1 and thus the connections '11' and '21' can be ignored. In the second layer, it is only concerned with connection '42' as the input value of x2 is 1 and thus 'the connections '12' and '32' can be ignored. In addition, the connection '22' is ignored as well because the connection '21' is already discarded and thus it is not worth to go through the connection '22' any more. As the basis of above descriptions, it indicates that it is not necessary to examine the whole network in order to find the rules that fire. In practice, it may significantly speed up the process of simulation when the corresponding rule set is generated by learning from Big Data.

4 Comparative Validation and Discussion

The authors have recently validated experimentally IEBRG against Prism [17] and Jmid-pruning against J-pruning and Jmax-pruning [19] in terms of classification accuracy and computational efficiency. They have also theoretically validated Rule Based Classification Networks against decision tree and linear list representations in terms of time complexity. With regards to classification accuracy, the authors use cross validation and check the overall accuracy, i.e. the proportion of correct classifications. With regards to computational efficiency in training stage, the authors check the number of rules and the average number of rule terms in order to reflect approximately the total number of iterations conducted during training stage. If a method generates more general and fewer rules, it indicates that the method needs less number of iterations and thus is more efficient in theory. In addition, the authors also check the time complexity using BigO notation to measure the computational efficiency in testing stage. If the complexity is lower, it indicates that the representation may make the predication on unseen instances perform more efficiently. For example, linear time is worse than logarithmic time in computational efficiency.

With regards to IEBRG, the authors conducted experiments on 10 datasets available from UCI repository [20]; they are Vote, Weather, Contact-lenses, Lense24, Breast-cancer, Nurse, Car, Lung-cancer, Kr-vs-kp and Iris. The experimental results show that IEBRG algorithm outperforms Prism in both accuracy and efficiency in most cases. In the classification accuracy, IEBRG performs a bit worse than Prism in one case (on Vote dataset) only. However, it even slightly outperforms Prism in three cases (on Nurse, Iris and Kr-vs-kp). In the computational efficiency, IEBRG generates more general and fewer rules in most cases. In three cases (on Lung-cancer, Nurse and Car datasets), IEBRG generates more rules than Prism. However, Prism discarded large number of rules in two of these cases (on Nurse and Car datasets). Therefore, it still shows Prism is computationally more expensive than IEBRG as discarded rules also need to conduct computation for their generation although they are eventually discarded.

With regards to Jmid-pruning, the authors conducted experiments on 10 UCI datasets namely, Vote, Weather, Contact-lenses, Lense24, Breast-cancer, Car, Lung-cancer, Iris, Segment and ionosphere. The experimental results show Jmid-pruning leads PrismTCS to perform a similar level of classification accuracy in comparison with J-pruning and Jmax-pruning in most cases but outperforms the two algorithms in some cases. With regards to efficiency, PrismTCS with Jmid-pruning may generate a rule set with similar level of rule complexity or even fewer but more general rules in comparison with J-pruning and Jmax-pruning. However, Jmid-pruning may perform better compared with Jmax-pruning in terms of computational efficiency. It can be seen by looking at the number of backward steps that Jmid-pruning needs a smaller number of iterations than Jmax-pruning to make Prism stop generating rules. Therefore, Jmid-pruning seems likely to be computationally more efficient when training data is very large.

With regards to Rule Based Classification Networks, the authors validated the representation theoretically using BigO notation. As mentioned above, the network representation could achieve that simulation process is run in divide and conquer search and the efficiency is O (log (n)). In contrast, list representation could only achieve a linear search process for the same purpose and the efficiency is O (n). For the purpose of predictive modelling, the network representation may contribute as many quicker decisions as possible in prediction stage in expert systems. The difference to listed rule representation in the efficiency can be significant when Big Data is used to generate a rule set.

As mentioned above, the authors' recent research is mainly concerned with accuracy and efficiency. The veracity is a measure of reliability leading to more accurate analyses and confident decision making as mentioned in Sect. 1. However, the accuracy can indicate the uncertainty existed in a model built based on a dataset. In addition, a data set may contain missing values or noise (incorrect records). Different strategies in dealing with the issues may lead to different predictive accuracy. In classification area, each algorithm may perform a particular level of tolerance to the presence of missing values or noise. As the basis of above descriptions, veracity is subject to data based modelling techniques in the authors' research. The higher level of predictive accuracy is more likely to introduce the higher degree to which the data can be trusted. In detail, rule generation method can provide a level of predictive accuracy and pruning algorithms may help improve the accuracy.

On the other hand, volume is a measure of data scalability leading to a particular level of computational efficiency. The data scalability could be reflected by its dimensionality, average number of attribute values and the number of instances. In the authors' research, pruning algorithms may speed up the process of modelling. The proper selection of model representations may speed up the process of simulation. Besides, the dimensionality issue can be resolved by using feature selection techniques such as entropy [18] and information gain [6] which are both based on information theory pre-measuring uncertainty present on data. In other words, it aims to remove those irrelevant attributes. When a dataset contains a large number of instances, it is possibly required to take advantage of sampling methods to choose those most representative instances. However, the authors have not yet taken feature selection and sampling into use in their current research but will do so further when large scale data is used.

5 Conclusions

This chapter has summarised the authors' more recent research in the area of rule based classification including generation, simplification and representation of classification rules. The authors have also introduced a unified framework for the construction of rule based classification systems by merging the three operations mentioned above systematically. The potential contribution to effective and efficient processing of Big Data has been discussed in the terms of volume and veracity. However, those validations are made theoretically or experimentally on some relatively small data in classification accuracy and computational efficiency. Therefore, the authors will further extend the validations onto large scale datasets and evaluate the novel methods more empirically in the concern of Big Data. They will also incorporate ensemble learning concepts and feature selection techniques with respects to the improvement of accuracy and efficiency in order to overcome the limitations that arise when Big Data is present and to make the approach more computationally intelligent.

References

1. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufman, Los Altos (1993)
2. Michalski, R.S.: On the quasi-minimal solution of the general covering problem. In: Proceedings of the Fifth International Symposium on Information Processing, Bled, Yugoslavia, pp. 125–128 (1969)
3. Cendrowska, J.: PRISM: an algorithm for inducing modular rules. Int. J. Man Mach. Stud. **27**, 349–370 (1987)
4. Bramer, M.A.: Automatic Induction of Classification Rules from Examples Using N-Prism, Research and Development in Intelligent Systems, vol. XVI, pp. 99–121. Springer, Cambridge (2000)
5. Bramer, M.A.: Using J-pruning to reduce overfitting of classification rules in noisy domains. In: Proceedings of 13th International Conference on Database and Expert Systems Applications—DEXA 2002, Aix-en-Provence, France, 2–6 Sept 2002
6. Bramer, M.A.: Principles of Data Mining. Springer, London (2007)
7. Smyth, P., Goodman, R.M.: Rule induction using information theory. In: Piatetsky-Shapiro, G., Frawley, W.J. (eds.) Knowledge Discovery in Databases, pp. 159–176. AAAI Press, California (1991)
8. Bramer, M.A.: Using J-pruning to reduce overfitting in classification trees. In: Research and Development in Intelligent Systems, vol. XVIII, pp. 25–38. Springer, Berlin (2002)
9. Stahl, F., Bramer, M.A.: Jmax-pruning: a facility for the information theoretic pruning of modular classification rules. Knowl. Based Syst. **29**, 12–19 (2012)
10. Stahl, F., Bramer, M.A.: Induction of modular classification rules: using Jmax-pruning. In: Thirtieth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, 14–16 Dec 2011
11. What is big data? http://www.sas.com/big-data/. 7 Dec 2013
12. Master data management for big data. http://www-01.ibm.com/software/data/infosphere/mdm-big-data/. 7 Dec 2013
13. Bramer, M.A.: Inducer: a public domain workbench for data mining. Int. J. Syst. Sci. **36**(14), 909–919 (2005)
14. Stahl, F., Bramer, M.A.: Computationally efficient induction of classification rules with the PMCRI and J-PMCRI frameworks. Knowl.-Based Syst. **35**, 49–63 (2012)
15. Bramer, M.A.: An information-theoretic approach to the pre-pruning of classification rules. In: Musen, M., Neumann, B., Studer, R. (eds.) Intelligent Information Processing, pp. 201–212. Kluwer, Dordrecht (2002)
16. Deng, X.: A covering-based algorithm for classification: PRISM. CS831: Knowledge discover in databases (2012)
17. Liu, H., Gegov, A.: Induction of modular classification rules by Information Entropy Based Rule Generation. In: V. Sgurev, R. Yager, J. Kacprzyk (Eds.) Innovative issues in intelligent systems. Springer, Berlin (in print)
18. Shannon, C.: A mathematical theory of communication. Bell Syst. Tech. J. **27**(3), 379–423 (1948). Fonn
19. Liu, H., Gegov, A., Stahl, F.: J-measure based hybrid pruning for complexity reduction in classification rules. WSEAS Trans. Syst. **12**(9), 433–446 (2013)
20. Bache, K., Lichman, M.: UCI Machine learning repository. http://archive.ics.uci.edu/ml. Irvine, CA: University of California, School of Information and Computer Science, 2013

Multi-granular Evaluation Model Through Fuzzy Random Regression to Improve Information Granularity

Nureize Arbaiy and Junzo Watada

Abstract Extracting new information through regression analysis is somewhat difficult in environment which contains fuzzy and random situation; shows simultaneous uncertainty. Given this coexistence of random and fuzzy information, the data cannot be adequately treated by a conventional regression method. Thus, in this paper, a fuzzy random regression is introduced to improve the extraction of weight of granules in a multi-granular decision making. The proposed model will manage the multi-granular linguistic labels provided by evaluators in order to compute collective assessments about the product samples that will be used by the decision maker to determine final decision. The proposed model is applied to oil palm fruit grading, as the quality inspection process for fruits requires a method to ensure product quality. We include simulation results and highlight the advantage of the proposed method in handling the existence of fuzzy random information.

Keywords Fuzzy regression · Multi-granular evaluation · Fuzzy random regression

1 Introduction

Granular computing studies a novel approach to computing system modeling and information processing. A detailed discussion of basic issues of granular computing is given in various papers [1–3, 19, 30–32]. The principles of the theory of

N. Arbaiy
Faculty of Computer Science and Information Technology,
University Tun Hussein Onn Malaysia, 86400 Johor, Malaysia
e-mail: nureize@uthm.edu.my

J. Watada (✉)
Graduate School of Information, Production and System,
Waseda University, 2-7 Hibikino, Wakamatsu, Kitakyushu 808-0135, Japan
e-mail: junzow@osb.att.ne.jp

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_11,

granularity have been applied in many studies [6, 7, 12]. Granules, levels, and relationships between them are the basic components of granular computing. Granules populate at a particular level. They are the subjects of investigation at that level. Different levels focus on different, though related, types of granules. The properties of granules collectively characterize a level of description and understanding. Levels are connected together through a partial order. Granules in different levels are related to each other [32]. In the granular modeling, the interaction between granules, levels and hierarchies may be represented by three the internal structure of a granule, the collective structure of all granules, and the overall structure of all levels. The three structures as a whole are referred to as granular structure [14]. Hence, when decision making structure or scheme uses this granules to evaluate a particular subject matter, then a multi-granules decision making scheme is formed. For example, several granules is used to decide certain output. In linear presentation, the evaluation might follows the following structure: $y = wg + e$ where y is the total evaluation, g is the granule or factor to decide y and w is the weight of the granule g, and error e.

Granule weighting establishes the importance of each granule relative to the others. Weighting factors play an essential role in many techniques for example to address multi-attribute decision making problem, such as simple additive weighting [11], the analytic hierarchy process [21], multi-attribute utility theory [8], the ordered weighted average [29], the Preference Ranking Organisation Method for Enrichment Evaluation [4], and Elimination and Choice Expressing Reality [20]. Multi-attribute evaluation also requires accurate weight information for each attribute. Appropriate weighting of the alternatives plays a pivotal role in multi-attribute evaluation, because the central aim of multi-attribute evaluation is to obtain the best alternative from among a set of evaluated alternatives. However, determining an attribute's weight is sometimes difficult if relevant data are either unavailable or difficult to obtain. Additionally the assignments of attribute weights may vary from one decision maker to another. Therefore, an appropriate method is required for determining these weights, as these decisions are crucial to the model's performance. Some method has been proposed and can be used to generate the attribute weight to alleviate the difficulties. For example, a regression analysis is one of the possible methods used to estimate the weights of the model [16, 23, 26].

A regression method analyzes statistical data to estimate the model coefficients in developing effective models. The conventional mathematical programming problem uses numerical deterministic values to these coefficients. In contrary, it is more realistic to take the estimated values of the coefficients as imprecise values rather than precise ones. In practical systems, probabilistic or/and vague situations include uncertain information such as predictions of future profits and incomplete historical data. Therefore, the mathematical programming models should be able to handle the above problems. That is, the above situations should be explicitly considered in the decision making process. For that reason, the fuzzy random regression model is introduced to solve such a problem with the existence of the randomness and fuzziness in historical data used for the approximation [28]. The property of fuzzy random regression model is used to allow for the co-existence of fuzziness and randomness in the data. Thus, the information of weight value which is deduced by the fuzzy random regression model is useful to build a decision making model.

Multi-granule problems can be dealt with by employing a regression model in which attributes, x_{jk}, are used to evaluate the total evaluation, y_j, and the relative importance of each attribute is given by coefficients, a_k. Fuzzy numbers are used instead of crisp numbers to describe the fuzzy information, and all of the observed values that express uncertainty in the system must be considered in the development of the model. Thus, the fuzzy regression model should contain all of the observed data within the estimated fuzzy numbers. However, the existing method of generating these weights for multi-granule problem is not handling the simultaneous occurrence of fuzzy random information yet such situation is obviously present in the real-world multi-granule evaluation. The weight that is produced only consider the fuzzy information, and neglects the inherent imprecision it its evaluation, although consideration of all inherent uncertainties is necessary in real-world decision-making.

The conceptual structure of a multi-granule decision-making contains a decision granule and it's weighting in which determining the relationship of several granules towards the total evaluation. A granule can be describe as a measurable quantity with a value that indicates the degree to which a particular objective is achieved. This granules plays a significant role to determine the total evaluation in the evaluation model structure which forming a multi-granular evaluation structure. A relevant measurement scale is used to assign value for this granule. On the other hand, the rating of each sample alternative is performed with respect to each granule and the weights given to each granule. Some of the multi-attribute structure is presented in hierarchy [15, 17, 21] which contains a collective evaluation defining levels of evaluation. The evaluation must therefore consider and satisfy all evaluation attributes. With that kind of theoretical background of evaluation structure, it makes the attributes or decision factor be a granule of information, and the weighting determines the relationship, that is multi-granular is characterized. The evaluation model were primarily developed for a crisp value environment and has been widely applied in various real-world application [5, 13, 18, 24]. As decision-making includes uncertain and vague information, fuzzy set theory was introduced to such decision making structure.

In light of the situation described above, mathematical programming models for decision support that consider the treatment of the inherent uncertainty associated with the model coefficients are necessary. Moreover, determining information granules and its relationship is important to construct an evaluation model. Therefore, the objective of this chapter study is to develop a multi-granular evaluation scheme which is able to generate the importance weight of decision attributes using the historical data that contain fuzzy random information and solves the multi-granular problem. In this study, the weight of information granular is deduced by means of fuzzy random regression method. Hence the fuzzy random regression is introduced to obtain the fundamental construct in building information granules.

The remainder of this paper is organized as follows. Section 2 introduces the fuzzy random regression for a multi-granular evaluation model. Section 3 describes the fuzzy random regression-based multi-granular evaluation model (FRR-MgEM). Section 4 explains on capturing the fuzziness and randomness in the data. Meanwhile, Section 5 presents a real application of the model in the evaluation of oil palm grading. Finally, conclusions are given in Section 6.

2 Multi-granular Through a Fuzzy Random Regression Approach

This section describes the development of fuzzy random based regression model to construct multi-granular evaluation model to solve multi-attribute problem. Fuzzy random variable is used to treat the presence of simultaneous fuzzy random data in multi-attribute decision-making.

The confidence intervals are expressed through the expectations and variances of fuzzy random variables. Fuzzy random data Y_j (output) and X_{jk} (input) for all $j = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, K$ are defined as $Y_j = \cup_{t=1}^{M_{Y_j}} \{ (Y_j^t, Y_j^{t,l}, Y_j^{t,r})_\Delta, p_j^t \}$ and $X_{jk} = \cup_{t=1}^{M_{X_{jk}}} \{ (X_j^t, X_j^{t,l}, X_j^{t,r})_\Delta, q_{jk}^t \}$, respectively. All fuzzy variables $(Y_j^t . Y_j^{t,l}, Y_j^{t.,r})_\Delta$ and $(X_j^t . X_j^{t,l}, X_j^{t.,r})_\Delta$ are obtained with probability p_j^t and q_{jk}^t for $j = 1, \ldots, N$, $k = 1, \ldots, K$ and $t = 1, \ldots, M$ or $t = 1, 2, \ldots, M_{X_{jk}}$, respectively.

In this study, the expectation is formulated based on an average of possibility (expresses a level of overlapping) and necessity (degree of inclusion) are defined based on [10]. The expected value of a fuzzy variable Y is presented as follows:

$$
\begin{aligned}
E[Y] = & \int_0^\infty \left(\frac{1}{2} \left[1 + \sup_{t \ge r} \mu_Y(t) - \sup_{t \le r} \mu_Y(t) \right] \right) dr \\
& - \int_{-\infty}^0 \left(\frac{1}{2} \left[1 + \sup_{t \le r} \mu_Y(t) - \sup_{t \ge r} \mu_Y(t) \right] \right) dr
\end{aligned}
\tag{1}
$$

where Y is assumed to be a fuzzy variable. From Eq. (1), the expected value of Y is defined as $E[Y] = \frac{a^l + 2c + a^r}{4}$.

Definition 1 Let X be a fuzzy random variable defined on a probability space $(\Omega, \sum, \Pr)$. The expected value of X is defined as

$$\begin{aligned} E[X] = \Bigg[\int_{\Omega} \int_{0}^{\infty} \left(\frac{1}{2} \left[1 + \sup_{t \geq r} \mu_{Z(\omega)}^{(t)} - \sup_{t < r} \mu_{Z(\omega)}^{(t)} \right] \right) dr \\ - \int_{\infty}^{0} \left(\frac{1}{2} \left[1 + \sup_{t \leq r} \mu_{Z(\omega)}^{(t)} - \sup_{t > r} \mu_{Z(\omega)}^{(t)} \right] \right) dr \Bigg] \Pr(d\omega) \end{aligned} \tag{2}$$

Definition 2 Let X be a fuzzy random variable defined on a probability space $(\Omega, \sum, \Pr)$ with expected value e. The variance of X is defined as

$$\mathrm{var[X] = E\Big\lfloor (X - e)^2 \Big\rfloor} \tag{3}$$

where $e = E[X]$ given by Definition 1.

Let us denote a fuzzy linear model with fuzzy coefficients $A_1^*, \ldots, A_K^*$ as $Y_j^* = A_1^* X_{j1} + \cdots + A_K^* X_{jK}$ where each Y_j^* denotes an estimate of the output and $A_k^{*r} = \left(\left[\frac{A_k^{*l} + A_k^{*r}}{2} \right], A_k^{*l}, A_k^{*r} \right)_{\Delta}$ are symmetric triangular fuzzy coefficients when triangular fuzzy random data X_{ik} are given for $j = 1, \ldots, N$, $k = 1, \ldots, K$.

The input data $X_{jk} = (x_{jk}, x_{jk}^l, x_{jk}^r)_{\Delta}$ and output data $Y_j = (y_j, y_j^l, y_j^r)_{\Delta}$ for $j = 1, \ldots, n$ and $k = 1, \ldots, K$, are fuzzy random variables. Therefore, the following relation should hold: $Y_j^* = A_j^* X_{j1} + \cdots + A_K^* X_{jK} \underset{FR}{\supset} Y_i$, $j = 1, \ldots, N$ where $\underset{FR}{\supset}$ is a fuzzy random inclusion relation. Thus, the one-sigma confidence interval that is induced by the expectation and variance of a fuzzy random variable is shown as follows:

$$I[e_X, \sigma_X] \triangleq [E(X) - \sqrt{var(X)}, E(X) + \sqrt{var(X)}] \tag{4}$$

Hence, the fuzzy random regression model with σ-confidence intervals is described as follows:

$$\begin{aligned} &\min_A \quad J(A) \sum_{k=1}^{K} (A_k^r - A_k^l) \\ &A_k^r \geq A_k^l \\ &Y_j^* = A_1^* I[e_{X_{j1}}, \sigma_{X_{j1}}] + \cdots + A_K^* I[e_{X_{jK}}, \sigma_{X_{jK}}] \underset{\tilde{h}}{\supset} I[e_{Y_j}, \sigma_{Y_j}] \\ &\quad j = 1, \ldots, N; \quad k = 1, \ldots, K \end{aligned} \tag{5}$$

Thus, the fuzzy random regression model with confidence intervals is given in the following expression:

$$Y_j = \sum_{j=1}^{m} A_j I[e_{X_{jk}} + \sigma_{X_{jk}}], \qquad j = 1, \ldots, N \tag{6}$$

The inclusion relation should be written as follows:

$$\begin{aligned} Y_j^r + [e_{X_{jk}} + \sigma_{X_{jk}}] &\leq \sum (A_K^r \cdot [e_{X_{jk}} + \sigma_{X_{jk}}])^T \\ Y_j^r - [e_{X_{jk}} - \sigma_{X_{jk}}] &\leq \sum (A_K^r \cdot [e_{X_{jk}} - \sigma_{X_{jk}}])^T \end{aligned} \tag{7}$$

The solution of the fuzzy random regression model with confidence interval can be rewritten as a problem of samples with one output and K input interval values [28].

3 A Fuzzy Random Regression-Based Multi-granular Evaluation Model

The new fuzzy random multi-granular model with a confidence interval is established using expectations and variances of fuzzy random variables. The proposed Fuzzy Random Regression-based Multi-granular Evaluation Model (FRR-MgEM) methodology is explained in the five stages as follows:

1. *Problem description.* The multi-granular evaluation model of this system consists of total evaluation, criteria, and alternatives to be evaluated. These model parameters can be deter-mined by examining the real-world problem model. The main objective is to select the good sample that shows a good quality among jth evaluated samples.
2. *Data elicitation.* Data are collected from the expert or from the historical data in the data base. The values for each criterion are assigned in a straightforward manner based on an intensity of importance scale [15]. Table 1 tabulates the intensity of importance scale.
3. *Data preparation for fuzzy random data.* The fuzzy random data are organized which result in the form of Y_j, X_{jK} for all $j = 1, \ldots, N$ and $k = 1, \ldots, K$. Y_j denotes the total evaluation of each sample j of alternatives, and X_{jK} represents the attributes. The fuzzy data are captured from the fuzzy intensity importance scale, and meanwhile the random data is captured from the difference of expert (decision maker) evaluation.
4. *Fuzzy random regression model to estimate the weight.* This steps is taken to estimate the weight of the granules considered in the evaluation.

Table 1 Intensity of importance scale [15]

1–3 Crisp value	Fuzzy value notation	Membership function $A = (a, h)$	Description
1	$\tilde{1}$	(1, 1)	Equal importance
2	$\tilde{2}$	(2, 1)	Equal to moderately importance
3	$\tilde{3}$	(3, 1)	Moderate importance
4	$\tilde{4}$	(4, 1)	Moderate to strong importance
5	$\tilde{5}$	(5, 1)	Strong importance
6	$\tilde{6}$	(6, 1)	Strong to very strong importance
7	$\tilde{7}$	(7, 1)	Very strong importance
8	$\tilde{8}$	(8, 1)	Very to extremely strong importance
9	$\tilde{9}$	(9, 1)	Extreme importance

a. Eliciting the confidence interval
The confidence interval of each fuzzy random variable is computed by inducing the expected value and variance of fuzzy random variable to construct the one-sigma confidence interval, $I[e_X, \sigma_X]$. The expected value $E[X]$ of triangular fuzzy random variable X and the variance of X is calculated accordingly.

b. Estimating the attribute's weight
Fuzzy random regression analysis is used to model the granules information of expert evaluation. Let A_k denote a granule for $k = 1, \ldots, K$ and Y_j^* is the total evaluation for $j = 1, \ldots, N$, where n is the number of candidate alternatives to be evaluated. A fuzzy random multi-granular evaluation model is described as Eq. (2).

5. *Decision-making and Analysis*. The weights $[\underline{a_K}, \overline{a_k}]$ are obtained from fuzzy random regression model. The estimated weight deduced by fuzzy random regression is used to calculate final score of evaluation for selecting the best samples among alternatives. Thus, multi-granular evaluation model is built. The total evaluation score is practical to rank the samples in order to find the best alternatives.

The solution method emphasizes on two important points. First, the weights of the granules are determined by fuzzy random regression method, which is to deal with the fuzziness and randomness observed in the data used to estimate the weight. Second, a multi-granular evaluation scheme that based on regression analysis is provided to assess the total evaluation. The analysis also provides the most appropriate alternative, a complete rank order of the alternatives, and an ordered list of the most excellent alternatives.

Table 2 Descriptive criteria for oil palm fruit grading

Criteria	Description
c_1: Color	Color of the fruitlet
c_2: Attached fruitlet	Number or percentage of attached fruitlet from the fruit bunch
c_3: Detached fruitlet	Number or percentage of detached fruitlet from the fruit bunch
c_4: Surface	External surface of the fruit bunch
c_5: Condition	Fruit bunch condition as a whole

4 Capturing Fuzziness and Randomness in the Data

An evaluation encompassing many inexact criteria is difficult to measure [9]. Such an evaluation is a challenging task due to its ambiguity and difficult formalization. However, if such imprecision is neglected, the formulated problem model may yield improper result. In real-world applications, statistical data may include both stochastic and fuzzy information at the same time. Fuzzy random variables can be explained by the use of a simple example. Assume that N evaluators are responsible for evaluating the jth product sample. Random-ness occurs because it is not known which response may be expected from any given respondent. In addition, fuzziness results if the observed response given by the respondent contains imprecision. Furthermore, if multiple decision makers are involved in evaluating the same alternatives or objectives, the differences in the decision makers' evaluations should also be considered. Such a situation may occur in real-life decision-making, and handling these types of data requires an appropriate approach. For these reasons, the observed statistical data may include both stochastic and fuzzy information, and thus, the decision-making analysis should provide an appropriate method of analysis to handle the presence of such hybrid uncertainty. Therefore, the combination of fuzziness and probability is important, and fuzzy random variables should be utilized as a basic tool for modeling optimization problems containing such uncertainties.

In this study of real case situation, the graders tend to judge a fruit based on their knowledge and preference by assigning numerical values or linguistic labels. These values are the degrees of satisfaction towards the total evaluation of the subject matter, in particular, fruit grade. Thus, the scores awarded by a grader are only approximations. Besides that, the evaluation given by the graders may contain imprecision such as, slightly defect, yellowish colour, few detached fruitlet, which is all contain fuzzy linguistic impression. Thus, to capture such fuzzy value from the grader, fuzzy importance scale (Table 2) is used. For example, grader may assign value 9 to certain decision attribute, which shows the information that the respective attribute has an extremely important towards a selection of quality fruit. This attribute value assignment is resembles the attribute rating in the multi-attribute evaluation scheme. All of the grader rating value for all respective decision attributes is collected for fruit samples and further be analyze.

Additionally, with presence of numerous graders that responsible to evaluate the fruit and provide the ratings, the difference of the graders evaluations also should be considered. Probability is used to calculate the proportional occurrence (probable situation) of the attribute rating. For example, one grader assigned fuzzy intensity 9, and another grader assigned fuzzy intensity 8, although evaluating the same samples of alternatives. Thus, such differences should account in the decision making process. Capturing the above-mentioned data, the observed statistical data may now include stochastic and fuzzy information, whereby the decision making process should provide an appropriate analysis.

5 Numerical Experiment

As oil-palm fruit grading describe above is a multi-attribute problem that exists in fuzzy random environment, the fuzzy random based evaluation scheme (FRR-MGEM) is applied. The main objective is to select the good quality of oil palm fruit bunches. The multi-granular evaluation is arranged in the four steps; (1) determine the model parameters, (2) rating the decision attributes, (3) perform fuzzy random analysis to obtain attribute weight, and (4) calculate the total score and perform aggregation to select quality fruit.

There are several attributes were used for oil palm fruit evaluations. In this case study we select the most frequent attributes suggested by the expert. Five attributes are selected during the inspection of quality process, denoted by where $k = 1,\ldots,5$, and is tabulated in Table 3. The general form of the oil-palm fruit evaluation is written as following regression model:

$$Y = [Y_j] = [C_1 x_{j1} + C_2 x_{j2} + \cdots + C_5 x_{j5}] = C x_j^t \tag{8}$$

where Y is the total evaluation of fruit quality, j is the number of samples, and C is the weight value that will be determined using historical data.

The values for each attribute were assigned by the grader using fuzzy importance scale, and the data were collected as listed in Table 4. For example, attribute c_1 of sample A_1 is assigned to 9, which represents the fact that colour has an extremely high importance for the selection of sample fruit. This means that in this case, the colour shows strong dominance over the other criteria. Apart from that, the random value is captured from the different value of rating provided by the graders. The frequency of occurrence of the rating for the particular attributes is calculated.

Data analysis is performed based on the collected data and is organised using fuzzy random data definition. The triangular fuzzy random input–output are pre-

Table 3 Data samples of oil palm fruit

Sample	(y_i, d_j)	c_1	c_2	c_3	c_4	c_5
A_1	(9, 0.2)	9	5	9	5	5
A_2	(9, 0.1)	9	5	8	6	6
A_3	(8, 0.2)	8	8	5	4	4
A_4	(5, 0.1)	3	8	4	4	5
A_5	(6, 0.1)	5	8	4	6	7
A_6	(7, 0.2)	6	5	8	3	6
A_7	(8, 0.2)	7	7	2	3	3
A_8	(8, 0.1)	7	6	3	3	2
A_9	(6, 0.1)	5	7	3	5	5
A_{10}	(5, 0.1)	5	5	7	6	8
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...

Table 4 Result by fuzzy random regression model regression model

Attribute	Weight		Weight	
	Lower	Upper	Central	Width
1	0.837	1.029	$a_1 = 0.933$	$h_1 = 0.096$
2	0.160	0.160	$a_2 = 0.160$	$h_2 = 0.000$
3	0.000	0.062	$a_3 = 0.031$	$h_3 = 0.031$
4	0.051	0.051	$a_4 = 0.051$	$h_4 = 0.000$
5	0.001	0.001	$a_5 = 0.001$	$h_5 = 0.000$

pared for fuzzy random regression computation to obtain the granule's weight and are given as follows:

[Input data]

$$
\begin{aligned}
&X_{11} = ((9,8,10)_{\Delta}, 0.5; (9,8,10)_{\Delta}, 0.5)\\
&X_{12} = ((5,4,6)_{\Delta}, 0.5; (6,5,7)_{\Delta}, 0.5)\\
&X_{13} = ((9,8,10)_{\Delta}, 0.5; (8,7,10)_{\Delta}, 0.5)\\
&X_{14} = ((5,4,6)_{\Delta}, 0.5; (6,5,7)_{\Delta}, 0.5)\\
&X_{15} = ((5,4,6)_{\Delta}, 0.5; (6,5,7)_{\Delta}, 0.5)\\
&X_{21} = ((8,7,9)_{\Delta}, 0.2; (7,6,8)_{\Delta}, 0.2; (7,6,8)_{\Delta}, 0.2; (7,6,8)_{\Delta}, 0.2; (7,6,8)_{\Delta}, 0.2)\\
&X_{22} = ((8,7,9)_{\Delta}, 0.2; (7,6,8)_{\Delta}, 0.2; (6,5,7)_{\Delta}, 0.2; (5,4,6)_{\Delta}, 0.2; (5,4,6)_{\Delta}, 0.2)\\
&X_{23} = ((5,4,6)_{\Delta}, 0.2; (2,1,3)_{\Delta}, 0.2; (3,2,4)_{\Delta}, 0.2; (7,6,8)_{\Delta}, 0.2; (8,7,9)_{\Delta}, 0.2)\\
&X_{24} = ((4,3,5)_{\Delta}, 0.2; (3,2,4)_{\Delta}, 0.2; (3,2,4)_{\Delta}, 0.2; (5,4,6)_{\Delta}, 0.2; (2,1,3)_{\Delta}, 0.2)\\
&X_{25} = ((4,3,5)_{\Delta}, 0.2; (3,2,4)_{\Delta}, 0.2; (2,1,3)_{\Delta}, 0.2; (3,2,4)_{\Delta}, 0.2; (8,7,9)_{\Delta}, 0.2)\\
&\ldots\\
&\ldots
\end{aligned}
$$

[Output data]

$$
\begin{aligned}
&Y_1 = ((9,8,10)_\Delta, 0.5; (9,8,10)_\Delta, 0.5)\\
&Y_2 = ((8,7,9)_\Delta, 0.2; (8,7,9)_\Delta, 0.2; (8,7,9)_\Delta, 0.2; (8,7,9)_\Delta, 0.2; (8,7,9)_\Delta, 0.2)\\
&Y_3 = ((7,6,8)_\Delta, 0.33; (7,6,8)_\Delta, 0.33; (7,6,8)_\Delta, 0.33)\\
&Y_4 = ((6,5,7)_\Delta, 0.25; (6,5,7)_\Delta, 0.25; (6,5,7)_\Delta, 0.25; (6,5,7)_\Delta, 0.25; (6,5,7)_\Delta, 0.25)\\
&Y_5 = ((5,4,6)_\Delta, 0.2; (5,4,6)_\Delta, 0.2; (5,4,6)_\Delta, 0.2; (5,4,6)_\Delta, 0.2; (5,4,6)_\Delta, 0.2)\\
&Y_6 = ((4,3,5)_\Delta, 1.0)
\end{aligned}
$$

The fuzzy regression problem with one-sigma confidence interval for data is $y_j = C_1 \cdot I[e_{x_{ji}}, \sigma_{x_{ji}}] + \cdots + C_1 \cdot I[e_{x_{ji}}, \sigma_{x_{ji}}]$ where $I[e_{x_{jk}}, \sigma_{x_{jk}}]$ for $k = 1, \ldots, 5$.

Based on fuzzy random regression model where $N = 120$ and $K = 5$, the model for determining the weight value for oil palm fruit is built as follows:

$$
\begin{aligned}
\min_A J(C) \quad &= \quad \sum_{k=1}^{S} (\overline{c_k} - \underline{c_k})\\
\textit{subject to} \quad & \quad \overline{c_K} \geq \underline{c_k}\\
& \quad y_j^r + (e_{Y_j} + \sigma_{Y_j}) \leq \sum_{k=1}^{S} \overline{c_k}(e \leq_{x_{j1}} + \sigma_{xj1})\\
& \quad y_j^r - (e_{Y_j} - \sigma_{Y_j}) \geq \sum_{k=1}^{S} \underline{c_k}(e \leq_{x_{j1}} - \sigma_{xj1})\\
& \quad j = 1, \ldots, 120;\ k = 1, \ldots, 5
\end{aligned}
\tag{9}
$$

The weights obtained from fuzzy random regression model (9) are shown in Table 5, where c_k and h_k denote a centre value and width of weight of attribute, respectively. Each weight is represented as $C_k = \langle c_k, h_k \rangle$, for $k = 1, \ldots, 5$. This result illustrates the weight for each attribute and shows the width of the decision, that is, the range of the evaluation. According to expert judgment, colour, attached fruitlets and detached fruitlets are the most important attributes, with weights of (0.933, 0.096), (0.160, 0.000) and (0.031, 0.031), respectively. This result is similar to the expert evaluation standards, where emphasis is placed on the colour, attached fruitlets and detached fruitlets attributes compared to other attributes. Other criteria of fruit characteristics were not strongly weighted. The data on detached fruitlets indicate that this attribute is also important, with a width value of 0.031. This result indicates that experts stress the detached fruitlets judgment. If, instead, the detached fruitlets showed a weak dominance, then the other criteria might represent strong dominance in the total evaluation.

The obtained result is used as input weights for evaluation ranking of oil palm fruit samples. The total scores of each sample were then computed. Table 6 partly tabulates the total evaluation for datasets obtained by fuzzy random based evaluation model and the evaluation obtained from the experts. The result shows the vagueness in expert judgment. From Table 6, we can see that the two alternatives A2 and A1 have similar evaluations, and they are not easily distinguishable via

Table 5 Comparison of evaluation by expert evaluation and FRR-MGEM model

Sample	Expert evaluation $\langle y_j, d_j \rangle$	FRR-MGEM total evaluation $\langle y_j, d_j \rangle$	Error
A_1	(9, 0.2)	(9.73, 1.14)	−0.73
A_2	(9, 0.1)	(9.75, 1.11)	−0.75
A_3	(8, 0.2)	(9.10, 0.92)	−1.10
A_4	(5, 0.1)	(4.41, 0.41)	−0.59
A_5	(6, 0.1)	(6.38, 0.60)	−0.38
A_6	(7, 0.2)	(6.80, 0.82)	0.20
A_7	(8, 0.2)	(7.87, 0.73)	0.13
A_8	(8, 0.1)	(7.74, 0.77)	0.26
A_9	(6, 0.1)	(6.13, 0.57)	−0.13
A_{10}	(5, 0.1)	(5.99, 0.70)	−0.99
...	...	...	...
...	...	...	...

Table 6 Comparison evaluation of fuzzy regression model (FR) and fuzzy random regression model (FRR)

Ranking	Sample	FR	Sample	FRR
1	A_1	(8.56, 077)	A_2	(9.75, 1.11)
2	A_2	(8.53, 0.84)	A_1	(9.73, 1.14)
3	A_3	(7.92, 0.83)	A_3	(9.10, 0.92)
4	A_7	(6.87, 0.67)	A_7	(7.86, 0.92)
5	A_{20}	(6.80, 0.60)	A_{11}	(7.80, 0.89)
6	A_8	(6.79, 0.61)	A_8	(7.73, 0.77)
7	A_{11}	(6.78, 0.73)	A_{20}	(7.68, 0.92)
8	A_6	(5.93, 0.64)	A_6	(6.79, 0.82)
9	A_{12}	(5.81, 0.59)	A_{16}	(6.71, 0.79)
10	A_{16}	(5.80, 0.69)	A_{12}	(6.64, 0.67)
...	...	...	...	...

FRR-MGEM, given their widths of 1.14 and 1.11, respectively, compared to the difference between the two evaluations of 9.75 versus 9.73. As such, the width data indicate that FRR-MGEM can play a pivotal role in data interpretation. Note that the result of the fuzzy random regression model had a wider width because of the consideration of the confidential interval. The width in this evaluation is important, as it reflects natural human judgment; a wider width indicates that the evaluation can captures more information under fuzzy judgments.

The regression model is commonly used to forecast the dependent variable *Y*, which is denoted as total evaluation in this study. Hence, this model is used to forecast the fruit grades, and it should achieve a higher accuracy to ensure the predicted value is as close as possible to the actual value. The smaller differences between the predicted value and the value actually observed shows that the model can provide highly accurate predictions. FRR-MGEM results −0.05 mean error.

The mean absolute error explains the average magnitude of errors in a set of forecasts, without considering error direction and FRR-MGEM yields 0.42. A residual plot is shown by scatter plotting, where the *x*-axis is the predicted value of *x* and the *y*-axis is the residual of *x*. In comparable with the Fuzzy Regression (FR) model, the accuracy measure of FRR-MGEM gain under forecasts as compared with the FR model.

6 Concluding Remarks

This case study allows us to draw the following conclusions:

- It is evident that fuzzy and random situations exist in the data, decision making, and the selection process environment, particularly discussed in this chapter is oil-palm fruit evaluation. Real-life problems may benefit from mathematical programming solutions. However, addressing the uncertainties that coexist in the decision-making environments considered here can be challenging. Different types of uncertainties may be generated by different circumstances, and it is important to address these uncertainties before the mathematical programming model is developed and solved.
- The differences in the decisions made by decision makers can be addressed by utilizing statistical analysis. That is, in this case study, fuzzy random variables are introduced in the multi-granular decision structure. We then employed expectations and variances of fuzzy random variables to construct confidence intervals of fuzzy random data, and built a fuzzy random regression model with confidence intervals for multi-attribute evaluation processes.
- The work described in this paper reveals that fuzzy evaluation in a multi-granular model can be used to better facilitate the decision-making process during the inspection of oil palm fruit quality. The regression analysis plays important role to determine important weight granular in the proposed multi-granular evaluation scheme. This enhancement is crucial to reduce the difficulties of determining the weight value during evaluation process, and at the same time handling the presence of fuzzy random information.

References

1. Bargiela, A., Pedrycz, W.: Granular Computing. Kluwer, Dordrecht (2002)
2. Bargiela, A., Pedrycz, W.: Recursive information granulation. IEEETrans. SMC-B **33**(1), 96–112 (2003)
3. Bargiela, A., Pedrycz, W.: Granular mappings. IEEE Trans. SMC A **35**(2), 292–297 (2005)
4. Brans, J.P., Vincke, Ph, Mareschal, B.: How to rank and how to select projects: the PROMETHEE method. J. Oper. Res. **24**(2), 228–238 (1986)

5. Cardoso, D.M., de Sousa, J.F.: A multi-attribute ranking solutions confirmation procedure. Ann. Oper. Res. **138**(1), 127–141 (2005)
6. de Andres, R., Garcia-Lapresta, J.L., Martinez, L.: A multi-granular linguistic model for management decision-making in performance appraisal. Soft. Comput. **14**(1), 21–34 (2010)
7. Herrera, F., Martínez, L.: A model based on linguistic 2-tuples for dealing with multigranularity hierarchical linguistic contexts in multiexpert decision-making. IEEE Trans. Syst. Man Cybern. B Cybern. **31**, 227–234 (2001)
8. Keeney, R.L., Raiffa, H.: Decisions with Multi-objectives. Wiley, New York (1976)
9. Li, Y., Chen, S., Nie, X.: Fuzzy pattern recognition approach to construction contractor selection export. Fuzzy Optim. Decis. Making **4**(2), 103–118 (2005)
10. Liu, B., Liu, Y.-K.: Expected value of fuzzy variable and fuzzy expected value models. IEEE Trans. Fuzzy Syst. **10**(4), 445–450 (2002)
11. Malczewski, J.: Propagation of errors in multicriteria location analysis: a case study. In: Fandel, G., Gal, T. (eds) Multiple Criteria Decision Making, Proceedings of the Twelfth International Conference, Hagen (Germany): 1995, Springer, Berlin, pp. 154–165 (1997)
12. MartLnez, L., Liu, J., Yang, J.B., Herrera, F.: A multi-granular hierarchical linguistic model for design evaluation based on safety and cost analysis. Int J Intell Syst **20**, 1161–1194 (2005)
13. Mavrotas, G., Diakoulaki, D., Capros, P.: Combined MCDA-IP approach for project selection in the electricity market. Ann. Oper. Res. **120**(1–4), 159–170 (2003)
14. Meiarov, Z.: Granular computing and its application in Rbf neural network with cloud activation function. J. Inf. Control Manage. Syst. **7**(1) (2009). Retrieved from http://kifri.fri.uniza.sk/ojs/index.php/JICMS/article/view/1032
15. Nureize, A., Watada, J.: A fuzzy regression approach to hierarchical evaluation model for oil palm grading. Fuzzy Optim. Decis. Making **9**(1), 105–122 (2010)
16. Nureize, A., Watada, J.: Multi-Attribute decision making in contractor selection under hybrid uncertainty. J. of Adv. Comput. Intelligence and Intell. Inf. **15**(4), 465–472 (2010)
17. Nureize, A., Watada, J.: Fuzzy random regression based multi-attribute evaluation and its application to oil palm fruit grading. Ann. of Oper. Res. pp. 1-17 (2011). Doi: 10.1007/s10479-011-0979-z10.1007/s10479-011-0979-zTI
18. Ogryczak, W.: Multiple criteria linear programming model for portfolio selection. Ann. Oper. Res. **97**(1), 143–162 (2000)
19. Pedrycz, W. (ed.): Granular Computing: an Emerging Paradigm. Physica-Verlag, Heidelberg (2001)
20. Roy, A.: Classement et choix en prsence de points de vue multiples (la mthode ELECTRE). la Revue d'Informatique et de Recherche Oprationelle (RIRO), Vol. 8, pp. 57–75 (1968)
21. Saaty, T.L.: The Analytic Hierarchy Process. McGraw-Hill, New York (1980)
22. Tanaka, H., Watada, J.: Possibilistic linear systems and their application to the linear regression model. Fuzzy Sets Syst. **27**(3), 275–289 (1988)
23. Tanaka, H., Hayashi, I., Watada, J.: Possibilistic linear regression for fuzzy data. Eur. J. of Oper. Res. **40**(3), 389–396 (1989)
24. Tavana, M., Sodenkamp, M.A., Suhl, L.: A soft multi-criteria decision analysis model with application to the European union enlargement. Ann. Oper. Res. **181**(1), 393–421 (2010)
25. Watada, J., Tanaka, H.: Fuzzy Quantification Methods. In: Proceedings of the 2nd IFSA Congress, at Tokyo, 66–69 (1987)
26. Watada, J.: Trend of fuzzy multivariant analysis in management engineering. In: Khosla, R. et al. (eds.) KES2005, LNAI 3682, Springer, Berlin. pp. 1283–1290 (2005)
27. Watada, J., Toyoura, Y.: Formulation of fuzzy switching auto-regression model. Int. J. Chaos Theory Appl. **7**(1, 2), 67–76 (2002)
28. Watada, J., Wang, S., Pedrycz, W.: Building confidence interval-based fuzzy random regression model. IEEE Trans. Fuzzy Syst. **11**(6), 1273–1283 (2009)
29. Yager, R.: On ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Trans. Syst. Man Cybern. **18**(1), 183–190 (1988)

30. Yao, Y.Y.: A partition model of granular computing. LNCS Trans. Rough Sets **1**, 232–253 (2004)
31. Yao, Y.Y.: Granular computing. Comput. Sci. (Ji Suan Ji Ke Xue) **31**, 1–5 (2004)
32. Yao, Y.Y.: Perspectives of Granular Computing Proceedings of 2005 IEEE International Conference on Granular Computing, 1, pp. 85–90 (2005)

Building Fuzzy Robust Regression Model Based on Granularity and Possibility Distribution

Yoshiyuki Yabuuchi and Junzo Watada

Abstract The characteristic of the fuzzy regression model is to enwrap all the given samples. The fuzzy regression model enables us to take the possibility interval for a granular instead of a single numerical value. This granular provides the wider treatment for us to human-centered understanding of the latent system. Such a granule or interval of fuzzy regression model is created by considering how far a sample is from the central values. That means when samples are widely scattered the size of a granular or an interval of the fuzzy model is widened. That is, the fuzziness of the fuzzy regression model is decided by the range of sample distribution. Therefore, outliers make the fuzzy regression model distorted. This chapter describes the model building of fuzzy robust regression from the perspective of granularity by removing improper data based on genetic algorithm. Moreover, let us build the fuzzy regression model that places the largest grade on the central point of scattering samples.

Keywords Fuzzy regression model · Robustness · Granularity · Possibility distribution · Human centered understanding · Outliers · Genetic algorithm

Y. Yabuuchi
Faculty of Economics, Shimonoseki City University,
2-1-1 Daigaku-Cho, Shimonoseki, Yamaguchi 751-8510, Japan
e-mail: yabuuchi@shimonoseki-cu.ac.jp

J. Watada (✉)
Graduate School of Information, Production and Systems, Waseda University,
2-4 Hibikino Wakamatsu, Kitakyushu, Fukuoka 808-0196, Japan
e-mail: junzow@osb.att.ne.jp

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_12,

1 Introduction

A fuzzy regression model is categorized into two types. The first type is an interval model based on the possibility concept and the second type is a non-interval model based on a least squares method. An interval model has been proposed by Tanaka [16, 17, 19] and a non-interval model has been proposed by Diamond [2, 3].

Tanaka et al. have proposed three models as an interval fuzzy regression model, a possibility model, a necessity model and a conjunction model. A difference in these three models is an inclusion relation between estimates $\mathbf{Y}$ of a fuzzy regression model and fuzzy interval data $\mathbf{y}$. The relation between a possibility model and observed data is written as $\mathbf{Y} \supseteq \mathbf{y}$, otherwise, the relation of a necessity model is $\mathbf{Y} \subseteq \mathbf{y}$, and the relation of a conjunction model is $\mathbf{Y} \cap \mathbf{y} \neq \phi$, respectively. An interval model is based on the concept of possibility theory, and this possibility model is known as the major model of a fuzzy regression model. A possibility model and a necessity model are referred to as an upper regression model and a lower regression model, respectively [5].

There are some ways to obtain an interval fuzzy regression model such as a least squares method [1, 4, 8, 12] and a linear programming (LP). This chapter, an interval model obtained by LP.

The objective of an interval fuzzy regression model is to describe a possibility of analyzed system, and to minimize a vagueness of a model in order to make us interpret a system with least bias. However, a possibility model describes a possibility of a system by enclosing data, a vagueness of a model is made bigger and a shape is made strained easily. Therefore, many possibility models have been proposed, which describe an essence of analyzed target although it is rough. These models have two approaches. One is to use an exponential possibility distribution which proposed by Tanaka et al. [5, 18]. And the other is to control the relation between a model and data, Ishibuchi and Tanaka [10], Yabuuchi and Watada [23, 26, 27] and so on have proposed.

In this chapter, the model which describes an essential possibility by controlling the relation between a model and data is focused on.

We have proposed two models. This chapter describes model building of fuzzy robust regression model with granule data or interval data by removing improper data based on genetic algorithm [14, 27]. Let us call this fuzzy robust model as the first model.

And, let us build the fuzzy regression model that places the largest grade on the central point of scattering samples [28–31] as the other model. This model is the second type model of fuzzy robust regression. Observed data are not real numbers as granular possibilities by the second type model. For this reason, the second type model has a less ill effect of irregular data than other models.

Granular concept enables us to understand a main essential target system [21] and to deal any data such as linguistic data [24].

The remaining is organized as follows: In Sect. 2, two conventional fuzzy regression models are introduced. In Sect. 3, an Asian environment, a relation between Asian economy and environment, are analyzed by a first type model of our fuzzy robust regression. In Sect. 4, a Japanese major rivers, and a relation between Asian economy and environment, is analyzed by a second type model of our fuzzy robust regression.

2 Fuzzy Regression Model

A fuzzy regression model is categorized into two types, a least squares model and an interval model. Therefore, in this section, two conventional fuzzy regression models, a least squares model by P. Diamond and an interval model by H. Tanaka, are introduced.

At first, a least squares fuzzy regression model by Diamond is illustrated. Then, an interval fuzzy regression model by Tanaka is introduced, this interval model is focused on this chapter.

2.1 Least Squares Fuzzy Regression Model

There are fuzzy regression models, a least squares model such as statistical regression model and an interval model to describe a possibility of analyzed target. A least squares fuzzy regression model has been proposed in order to give error estimates in the form of residuals by Diamond [2, 3]. Therefore, this model is focused on a linear least squares estimation for vague data.

Observed data $(\mathbf{x}_i, \mathbf{y}_i), (i = 1, 2, \ldots, n)$ are triangular fuzzy numbers, where p dimensional explanatory variables $\mathbf{x}_i = \left(\mathbf{x}_i^L,\, \mathbf{x}_i^C,\, \mathbf{x}_i^U\right)$ and a dimensional response variable $\mathbf{y}_i = \left(y_i^L, y_i^C, y_i^U\right)$ are used. Here, L, C, and U denote the lower limit, the center, and the upper limit of a fuzzy number in this paper, respectively. We assume $\mathbf{x}$ are random variables and $\mathbf{x}_i^L \geq 0$. Although a least squares fuzzy regression model by P. Diamond has p dimensional real-valued coefficients vector $\mathbf{b}$, outputs vector $\mathbf{Y}$ are triangular fuzzy numbers because $\mathbf{x}$ are triangular fuzzy numbers.

Then, parameters $\mathbf{b}$, which give a minimal distance between $\mathbf{y}$ and $\mathbf{Y}$, are regression coefficients of the best fit model.

That is, the coefficient of a least squares fuzzy regression model is obtained by minimizing the Eq. (1).

$$\sum_{i=1}^{n}\left\{\left(y_i^C - Y_i^C\right)^2 + \left(y_i^L - Y_i^L\right)^2 + \left(y_i^U - Y_i^U\right)^2\right\} \tag{1}$$

2.2 Interval Fuzzy Regression Model

As observed data should embody possibilities that a considered system has, the measured data can be interpreted as the possibilities of the system. Therefore, a fuzzy regression model is built in terms of the possibility and evaluates all observed values as possibilities that the system should contain. In other words, the fuzzy regression model aims to be built so that it could contain all observed data in the estimated fuzzy numbers resulted from the model. Therefore, this model is applied to many applications [11, 14, 20, 22].

The fuzzy regression equation is written as in the following:

$$\begin{aligned} Y_j &= A_1 x_{1j} + \cdots + A_p x_{pj} = \mathbf{A}\mathbf{x}_j, \\ x_{1j} &= 1; j = 1, 2, \ldots, n, \end{aligned} \tag{2}$$

where each regression coefficient A_i is a symmetric triangular fuzzy number $A_i = (a_i, c_i)$ with center a_i and width c_i. In the Eq. (2), x_j denotes the jth data. A various type of a fuzzy coefficient is used, a symmetric triangular shaped fuzzy number is used as fuzzy coefficients in this chapter.

According the extension principle,

$$Y_j = \mathbf{A}\mathbf{x}_j = (\mathbf{a}, \mathbf{c})\mathbf{x}_j = \left(\mathbf{a}\mathbf{x}_j, \mathbf{c}\left|\mathbf{x}_j\right|\right), \tag{3}$$

where $\left|\mathbf{x}_j\right| = \left[\left|x_{1j}\right|, \left|x_{2j}\right|, \ldots, \left|x_{pj}\right|\right]$.

The output of fuzzy regression Eq. (2), whose coefficients are fuzzy numbers, results in a fuzzy number.

The regression model with fuzzy coefficients can be expressed with center $\mathbf{a}\mathbf{x}_j$ and width $\mathbf{c}\left|\mathbf{x}_j\right|$. When sample $\left(\mathbf{y}_j, \mathbf{x}_j\right)(j = 1, 2, \ldots, n)$ with center y_j and width d_j is given as fuzzy number $\mathbf{y}_j = \left(y_j, d_j\right)$, the inclusion relation between the model and the data should be hold as follows:

$$\begin{aligned} \mathbf{a}\mathbf{x}_j + \mathbf{c}\left|\mathbf{x}_j\right| \geq \mathbf{y}_j, \\ \mathbf{a}\mathbf{x}_j - \mathbf{c}\left|\mathbf{x}_j\right| \leq \mathbf{y}_j, \end{aligned} \tag{4}$$

In other words, the possibilistic regression model is built to contain all data in the model. When the width of the model is large, the expression of its regression equation is vague. It is better and more convenient to obtain a clear and rigid expression. Therefore, the width of the regression should be minimized as removing the vagueness of the model as possible. The fuzzy regression model is formulated to minimize its width under constraints (4). This problem results in a linear programming.

Using the notations of observed data $\left(\mathbf{y}_j, \mathbf{x}_j\right)$,

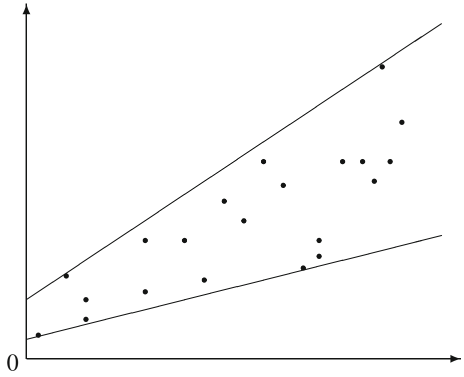


Fig. 1 Fuzzy regression model

$$\mathbf{y}_j = \left(y_j, d_j\right), \mathbf{x}_j = \left[x_{1j}, x_{2j}, \ldots, x_{pj}\right] (j = 1, 2, \ldots, n),$$

fuzzy coefficients $(\mathbf{a}, \mathbf{c})$ of the regression model can be mathematically written in the following LP problem:

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{n} c\left|x_j\right| \\ \text{subject to} & \\ & \mathbf{a}\mathbf{x}_j + \mathbf{c}\left|\mathbf{x}_j\right| \geq \mathbf{y}_j \\ & \mathbf{a}\mathbf{x}_j - \mathbf{c}\left|\mathbf{x}_j\right| \leq \mathbf{y}_j \\ & \mathbf{c} \geq \mathbf{0}\, (j = 1, 2, \ldots, n). \end{array} \tag{5}$$

Solving the LP problem mentioned above, we have the possibilistic regression shown in Fig. 1. Relation (4) between the model and the data is held as shown in Fig. 1. This fuzzy regression contains all data in its width and results in expressing all possibilities that data embody and the considered system should have. It is possible in the formulation of the fuzzy regression model to treat non-fuzzy data with no width by setting width d to 0 in the above equations.

3 First Type Model of Fuzzy Robust Regression

An interval fuzzy regression model describes a possibility of analyzed system, and the model is built so as to minimize its ambiguity, as mentioned above. However, a vagueness of a model is made bigger and a shape is made strained easily. A fuzzy robust regression model is one of interval fuzzy regression models, takes into account the distortion problem of a model shape, and aims to describe an essence of analyzed target.

It is possible to consider the data is granular, and is observed as fuzzy numbers or real numbers. Therefore, regardless of the state of the data, a fuzzy regression model is intended to describe accurately the possibility of the system. Our fuzzy robust regression model has two approaches to describe an essence of analyzed target. The first approach employs the concept of a conjunction model to treat granule data distorting a model shape, this model is the first type model of fuzzy robust regression. The second approach employs a maximization the total possibility grade from a model and data, this model is the second type model of fuzzy robust regression.

3.1 Formulation of First Type Model

Irregular values are often given in real world problems. These irregular values come from such causes as the errors of observation methods, the miss-reading of the observed values, the miss-behavior of observation instruments, and so on. Such data distort the expression of the possibility that the latent system should have. Therefore, the influence of the irregular data on the system must be removed in building a regression model or be controlled.

The concept of distance is employed to build a fuzzy robust regression model and to remove the influence of irregular data in building the model.

Let us consider the fuzzy regression shown in Fig. 1 that expresses properly the possibilities of the considered system. When an irregular sample denoted as ⊙ happens to be mixed in the data, model (2) is distorted largely from the proper figure of the possibility that the considered system should have, as shown in Fig. 2. As illustrated in figures, such irregular data influence on the fuzzy regression model very much. Furthermore, we should discriminate between the error and the possibility included in the data to build a possibility model [26]. This is an approach to build a fuzzy robust regression model by granule data or fuzzy data.

We evaluate both the error distance of data from the model and the fuzziness derived from the system separately in terms of the concept of distance. That is, in building the model we minimize not only the fuzziness included in the model but also the error distance of the samples from the model. This fuzzy regression of data with error is formulated to minimize not only its fuzzy width of the model but also the distance of irregular samples from the model.

As a result, the error between *j*th sample and the possibilistic model, that is, the distance, r_j, between *j*th sample and the possibilistic model can be written as follows:

$$r_j = \begin{cases} y_j - Y_j^U; & Y_j^U \le y_j \\ 0; & Y_j^L \le y_j \le Y_j^U \\ Y_j^L - y_j; & y_j \le Y_j^L \end{cases} \tag{6}$$

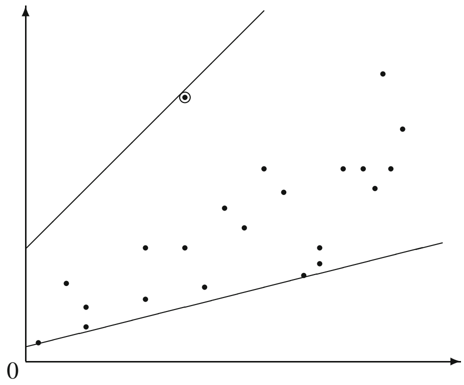


Fig. 2 Fuzzy regression model based on data with outlier⊙

where Y_j^U and Y_j^L denote the upper and lower boundaries of the model, respectively. Let us define the evaluation function of the fuzzy robust regression model using a distance function (6) as follows:

$$J = \sum_{j=1}^{n} r_j + K \sum_{i=1}^{p} c_i, \tag{7}$$

where constant K in the evaluation function is a positive real value that will be decided as a parameter how much to weigh the possibility included in error data.

Let us note that the better term of the above mentioned evaluation function can not be written only in $\sum_{i=1}^{p} c_i$, but also in several expressions and we should define it according to real problems or its objective as same as in the conventional fuzzy regression model.

Parameter K can define the property of the model and has the following meaning. When K is taken as a small value, the model results in the conventional fuzzy regression model because it emphasizes on minimizing the error distance against the model rather than the width of the model. On the other hand, when K is taken as a large value, the model results in the fuzzy regression model without error data because it emphasizes on minimizing the width of the model rather than the error distance against the model. When K is taken as a sufficiently large value the model comes in as a similar model to a statistical one. Using this parameter K it is possible to reflect in model building the knowledge which decision makers or analyzers have obtained from experience of model building.

We cannot tell which data are normal or which are error data, irregular samples or outliers. The fuzzy robust regression analysis discriminates whether data are normal or irregular based on the concept of distance.

The problem to obtain the best fuzzy regression model without the influence of error data in the possible combination of error data results in a combinatorial optimization problem. A genetic algorithm is employed to solve this combinatorial optimization problem.

As mentioned above, the approach to handle data with error was discussed. However, when errors including in data are viewed as a fuzziness, a vagueness and so on, this approach us to handle granule data or fuzzy data.

3.2 Elimination of Irregular Data

When n samples are given, it should be required to calculate 2^n times LP problems in constructing the above mentioned robust model. Generally, it requires the huge numbers of combinatorial calculations to obtain a fuzzy regression model. Nevertheless, the number of feasible solutions is very limited comparing with the number of these calculations 2^n.

All given samples should be treated in the sense of possibility in the fuzzy regression model. In other words, all the given samples should be included in the possibility that the fuzzy regression model expresses. This means that the shape of the fuzzy regression model is determined by the samples at the marginal boundaries of the model. On the other hand, samples that are distributed in the inside and central portion do not have any influence on determining the shape of the model. Therefore, in building the fuzzy robust regression model, out of samples that are near on the marginal boundaries, we can find irregular samples which should not be interpreted in terms of the possibility of system. If we can eliminate such irregular samples, the fuzzy robust model can be effectively and efficiently built. In this paper we employ a hyperelliptic function in order to detect irregular samples that might be on or near to the marginal boundaries.

If we can cluster some portion of samples that should be included in the possibility of the system using a hyperelliptic function, the combinatorial calculations can be reduced into the combination of the remained samples that are not included in the cluster. When h samples are selected out of the total n samples using the hyperelliptic function, the combinatorial calculation can be reduced from 2^n to 2^h.

3.3 Analysis of Asian Economy and Environment

It is widely known that an expansion of economic activity brings about an increase in population and energy consumption, and the increase in population and energy consumption results in an environmental change such as a large amount of air pollutant which influences the environment. Since vital economic activity is pursued in Asian region, Asian region is worried about that her energy consumption causes in an environmental change.

In this section, we analyze the relation between economic activity and environmental change in the Asian region. Population, GDP and an amount of primary energy consumption are employed for denoting economic activity, NO_x, SO_x and CO_2 are employed for describing environmental influence.

A primary energy consumption used to consist of commercial energies as coal, oil, gas and electric, but we also include plant energy in the primary energy consumption because many developing countries always depend on plant energy that is non-commercial energy in quit large amount.

SO_x and NO_x relate to global acidification and CO_2 relates to the greenhouse effect. NO_x relates to the formation of photochemical smog whose main component is ozone, ozone brings about destruction of plants and the greenhouse effect. Therefore, we employ NO_x, SO_x and CO_2 as objective variables in analyzing the environment.

As being analyzed in this section seem not to have large error. On the conventional fuzzy regression model, the width of the fuzzy regression is large. Therefore, we employ our first type model of fuzzy robust regression [11, 20, 23, 26, 27] to analyze this environmental change without influence of some countries.

Data [13] are expressed in terms of a natural logarithm and are observed on 1975.

As inputs, population is denoted by X_1, GDP by X_2 and a primary energy consumption by X_3, as outputs, estimations of NO_x by Y_{NO_x}, SO_x by Y_{SO_x} and CO_2 by Y_{CO_2}, respectively.

In this section, our purpose is to analyze the possibility of the relation between economic activity and environment by our first type model of fuzzy robust regression. We intend to analyze the tendency in this paper, so we do not consider collinearity. In this analysis, we employ a regression equation as follows:

$$Y = A_0 + A_1X_1 + A_2X_2 + A_3X_3.$$

At first, let us confirm statistical regression models based on a least squares method, denoting estimations of NO_x by $Y^{LS}_{NO_x}$, SO_x by $Y^{LS}_{SO_x}$ and CO_2 by $Y^{LS}_{CO_2}$, respectively. Then, these models are as follows:

$$\begin{aligned} Y^{LS}_{NO_x} &= 2.539 - 0.162X_1 - 0.176X_2 + 1.287X_3, \\ Y^{LS}_{SO_x} &= 2.362 - 0.500X_1 - 0.183X_2 + 1.875X_3, \\ Y^{LS}_{CO_2} &= 6.101 - 0.466X_1 + 0.150X_2 + 1.450X_3. \end{aligned}$$

These partial regression coefficients of three models makes us be understood increasing in a primary energy consumption, X_3, leads to an increase of environmental factors, NO_x, SO_x and CO_2, and a primary energy consumption is the most influential environmental factor than other factors. However, population, X_1, and environmental factors are inversely related. In addition, growth of GDP, X_2, leads decrease of NO_x and SO_x, the increase in CO_2.

Therefore, it was confirmed that it is possible to suppress the deterioration of the environment by reducing a primary energy consumption.

Next, conventional fuzzy regression models are obtained as follows:

$$
\begin{aligned}
Y_{NO_x}^{CM} &= (2.810, 0.11) + (0, 0.309)X_1 + (0, 0)X_2 + (0.922, 0)X_3,\\
Y_{SO_x}^{CM} &= (2.799, 0) + (0, 1.064)X_1 + (0, 0)X_2 + (1.136, 0)X_3,\\
Y_{CO_2}^{CM} &= (6.657, 0) + (0, 0.956)X_1 + (0.236, 0)X_2 + (0.171, 0)X_3.
\end{aligned}
$$

As well as statistical regression model, in fuzzy regression models, primary energy consumption has a positive coefficients. However, the impact of primary energy consumption is high, the other coefficients has a small value. A coefficient of GDP is 0, in NO_x model and SO_x model. Therefore, we can interpret these fuzzy regression models does not be describing the relation between the Asian environment change and an economic activity. The system possibility between the Asian environment and an economic activity is distorted in the conventional fuzzy regression model. Data used in this analysis are a real-valued, it can be considered a point in the granular sample has been observed. For data that distorting the model, as in the conjunction model, our first type fuzzy regression model describes the portion of the possibility data have.

Eight countries, which might distort a conventional fuzzy regression model, should be eliminated by a hyperellipse as distorting the possibility of the system. We analyzed objective system by using a distance function (6) to these eight countries and enclosing the other countries. The hyperellipse of NO_x is as follows:

$$
\begin{aligned}
&23.389(Y_{NO_x} - 4.589)^2 + 7.591(Y_{NO_x} - 4.589)(X_1 - 3.171) + \cdots\\
&\quad - 17.116(X_2 - 2.697)(X_3 - 2.362) + 44.296(X_3 - 2.362)^2 = 3.545,
\end{aligned}
$$

the hyperellipse of SO_x is as follows:

$$
\begin{aligned}
&1.322(Y_{SO_x} - 4.712)^2 + 1.322(Y_{SO_x} - 4.712)(X_1 - 3.171) + \cdots\\
&\quad - 7.399(X_2 - 2.697)(X_3 - 2.362) + 10.193(X_3 - 2.362)^2 = 3.226,
\end{aligned}
$$

and the hyperellipse of CO_2 is as follows:

$$
\begin{aligned}
&3.183(Y_{CO_2} - 8.451)^2 + 2.969(Y_{CO_2} - 8.451)(X_1 - 3.171) + \cdots\\
&\quad - 5.105(X_2 - 2.697)(X_3 - 2.362) + 12.234(X_3 - 2.362)^2 = 3.203.
\end{aligned}
$$

On the first type model of fuzzy robust regression, the number of combinations which eight countries are treated by possibility concept or distance concept is 2^8, genetic algorithm is employed to search a fuzzy robust regression model over all combination. We set population size to 200, crossover rate to 70 %, mutations rate to 1 % and production rate to 90 % as parameters of the genetic algorithm. In Table 1, a search rate of SO_x model ($K = 1$) and CO_2 model ($K = 1$) are low, and each rate are 36 % and 46 %. The circumstance of terminations is shown in

Table 1 Searching results by genetic algorithm

	K	Average of terminate generation	Average of solution's fitness	Average of generates	Terminal rate of optimum solution
NO_x	1	18	0.228	838	92
	100	18	8.298	854	100
SO_x	1	19	0.717	928	36
	100	25	26.934	903	94
CO_2	1	22	0.739	955	46
	100	19	12.581	954	90

Table 2 Searching rate of SO_x and CO_2 (K = 1) by genetic algorithm

Model	SO_x		CO_2	
	Fitness	Terminate rate	Fitness	Terminate rate
Optimum	0.717176	36	0.738750	46
Second optimum	0.717177	56	0.738752	90
Third optimum	0.717178	98	0.738753	100

Table 2. It should be noted that the optimum model could not be surely obtained because the genetic algorithm searches a model randomly based on probability. Therefore, we search a model 50 times under the same condition. The rate of obtaining the optimum model in the case SO_x and CO_2 ($K = 1$) are lower than the rate in the case SO_x and CO_2 ($K = 100$), but the second optimum model or the third optimum model in the case SO_x and CO_2 ($K = 1$) are gotten with high probability. The optimum, the second optimum and the third optimum model in the case SO_x and CO_2 ($K = 1$) have small deference between each fitness and model, and we can regard each model as the same model. It is not a serious problem that the rate of obtaining the optimum model is low because the difference between the optimum model and the obtained model of SO_x ($K = 1$) and CO_2 ($K = 1$) is very small. The optimum model is shown in Eqs. (9), (11) and (13).

$$\begin{aligned} Y_{NO_x}^{K=1} &= (2.717, 0.107) + (-0.382, 0.082)X_1 \\ &\quad + (-0.290, 0.040)X_2 + (1.543, 0)X_3 \end{aligned} \tag{8}$$

$$\begin{aligned} Y_{NO_x}^{K=100} &= (2.830, 0) + (-0.363, 0.071)X_1 \\ &\quad + (-0.214, 0)X_2 + (1.475, 0)X_3 \end{aligned} \tag{9}$$

NO_x model $Y_{NO_x}^{K=1}$ with $K = 1$, (8), has a small width. Coefficient of X_1 and X_2 has a small width, and these width are 0.082 and 0.040.

NO_x model does not have a large difference between the cases $K = 1$ and $K = 100$. This is common to the models of NO_x, SO_x and CO_2, the Asian population is scattered large. In comparison between the statistical model and the optimum model (9), there is a sharpness in the coefficients of $Y_{NO_x}^{K=100}$, a population and GDP have a large degree of inverse proportion to NO$_x$. In addition this, a primary energy consumption increase lead to a large increase in NO$_x$ observation.

$$\begin{aligned} Y_{SO_x}^{K=1} = {} & (2.537, 0) + (-1.470, 0.269)X_1 \\ & + (-0.583, 0.448)X_2 + (3.530, 0)X_3 \end{aligned} \tag{10}$$

$$\begin{aligned} Y_{SO_x}^{K=100} = {} & (1.979, 0) + (-0.578, 0.209)X_1 \\ & + (0.164, 0.000)X_2 + (1.936, 0)X_3 \end{aligned} \tag{11}$$

On Eqs. (10) ($K = 1$) and (11) ($K = 100$), the coefficient of primary energy consumption (X_3) of SO$_x$ model is a positive value with no width and the Eq. (10) is about one third of the Eq. (11). However the coefficient of X_2 ($K = 100$) is a positive value with almost no width and about one-third in the absolute value of it.

The difference between the model $Y_{SO_x}^{K=100}$ and the statistical model, is that the GDP is turned to SO$_x$ observed loss in the model $Y_{SO_x}^{K=100}$. This can be understood subjectively.

$$\begin{aligned} Y_{CO_2}^{K=1} = {} & (6.042, 0.394) + (-1.101, 0.264)X_1 \\ & + (0.196, 0.081)X_2 + (2.162, 0)X_3 \end{aligned} \tag{12}$$

$$\begin{aligned} Y_{CO_2}^{K=100} = {} & (5.960, 0) + (-0.403, 0.087)X_1 \\ & + (0.349, 0)X_2 + (1.202, 0)X_3 \end{aligned} \tag{13}$$

CO$_2$ model has a large different between $K = 1$ model $Y_{CO_2}^{K=1}$ (12) and $K = 100$ model $Y_{CO_2}^{K=100}$ (13). $K = 100$ model has small width of a coefficient of X_1 and X_2. Several countries have a different tendency on X_1 and X_2. The coefficient of X_1 is about one third of Eq. (12), the center of the coefficient of X_2 is about half of Eq. (12), respectively.

Difference between the model $Y_{CO_x}^{K=100}$ and the statistical model, is that the GDP is to double the amount of CO$_2$ observed in the model $Y_{CO_2}^{K=100}$. This can be understood subjectively, too.

This fact, the proposed model is describing the essence of the system, realize assent subjective. In the context of an economic activity and the environment, eight countries distinctive rather than have been removed at the time of model building, which was used only some of the features.

As the result of analyzing, eight countries distinctive as follows:

Japan: Japan employs a measure for the reduction of the emission of air pollutants, and the Environmental Agency monitors the amount of air pollutant. Each of NO$_x$, SO$_x$ and CO$_2$ are the smallest amount of the emission in Asian countries.

Indonesia: A number of farm workers is a half of all workers and agricultural product shares a quarter of GDP. A crude oil, a natural gas and a petrochemical share three fourth of the export. The amount of the emission of SO_x is smaller than another countries.

Taiwan: Economy is mainly industrial to manufacture industrial products such as electric devices. The amount of the emission of NO_x is smaller than another Asian countries.

Vietnam: Since Vietnam War has ended when data are observed, Vietnam can not make own supply. A fossil fuel is mainly used, and the amount of the emission of CO_2 is larger than another Asian countries.

Singapore: The main industry is to make a refined product from an imported crude oil. The amount of the emission of NO_x is smaller than another Asian country and CO_2 are larger.

Nepal: Nepal is an agricultural country, and 90 % of working force are a farm worker. The amount of the emission of NO_x, SO_x and CO_2 are smaller than another Asian countries.

Mongolia: Mongolia is an agricultural country based on pasturage. Mongolia is remarkably developing an industry. The amount of the emission of SO_x is smaller, NO_x and CO_2 are larger than another Asian countries.

Brunei: The export is crude petroleum and natural gas. The amount of the emission of NO_x and CO_x are smaller than another Asian countries.

We have discussed about eight countries that locate marginal of possibility area in case of $K = 100$ as follows. Small amount of the emission of NO_x is observed in Japan, Taiwan, Singapore, Nepal and Brunei, on the other hand, large amount of the emission of NO_x is observed in Mongolia. Small amount of the emission of SO_x is observed in Japan, Indonesia, Nepal and Mongolia. Small amount of the emission of CO_2 is observed in Japan, Nepal and Brunei, Large amount of the emission of CO_2 is observed in Vietnam, Singapore and Mongolia. Japan is enforcing the Environmental Pollution Prevention Act, Nepal is an agricultural country and has small amount of the emission of pollutant because Nepal does not have a lively industry makes pollutants (NO_x, SO_x and CO_2). Since Mongolia is an agricultural country, which mainly based on pasturage and is developing industry, the result of analysis is like as above. The main industry in Singapore is to make a refined product from an imported crude oil. Therefore, large amount of the emission of NO_x is observed in Singapore.

In Asian region, in the case that GDP and consumption of primary energy are large value, the amount of emission of CO_2 and SO_x is large scale. On the other hand, in the case that GDP are small value and consumption of primary energy are large value, the amount of emission of NO_x is large scale. Moreover, the larger population is, the smaller the amount of the emission of CO_2, SO_x and NO_x becomes. Since coefficients of primary energy consumption have a large value than another coefficients, we can understand that primary energy consumption and emission of pollutants relate strongly with each other.

As a result of economic activity, GDP is expressed. In developing country, GDP has concerned with manufacture industry, and manufacture industry uses much energy that is mainly primary energy. By consuming primary energy, air pollutants are made. A country which economic activity is developing increase in population, people mainly use plant energy in developing country. Therefore, a growth of GDP brings an increase of population, and an increase of population brings a large amount of the emission of air pollutants.

In this analysis, we employed population, GDP and primary energy consumption as input parameters to analyze the economic activity and the environment of Asian region.

In this analysis, we can have understood the tendency of Asian country and the relation between economic activity and environment by our first type model of fuzzy robust regression.

4 Second Type Model of Fuzzy Robust Regression

In the first type model of fuzzy robust regression, the concept of a conjunction model to treat granule data distorting a model shape and the possibility concept are employing the other data in order to describe an essence of analyzed target. The sum of distance between a first type model and observed values is minimized in the concept of a conjunction model, the ambiguity of the model is minimized in the possibility concept.

The second type model of fuzzy robust regression maximize the total possibility grade from a model and data in order to describe the possibility distribution of analyzed target by the model. For this reason, the center of the second type model coincides with the center of the possibility distribution.

It is possible to obtain the model without an inclusion relation between data and model. Model without using an inclusion relationship may describe the essence of the system. Moreover, this model less sensitive to outliers, and a shape of this model is less distortion.

4.1 Formulation of Second Type Model

In a interval fuzzy regression model, the possibility is represented by an interval so that the interval includes the whole data observed from the focal system [17, 19]. It is most characteristic that samples influence and distort the shape of the model, if samples are separated far from the center of data [10, 26].

On the other hand, the pivotal role of the center position of the system is emphasized in building a possibilistic regression model instead of employing an interval to describe the possibility of a focal system. Tanaka and Guo [19] employ exponential possibility distribution to build a model, while Inuiguchi et al. [9], Tajima [15] and Yabuuchi and Watada [28–31] are working independently on coinciding between the centers of possibility distribution and the center of a possibilistic regression model.

Yabuuchi and Watada proposed the model to describe a system possibility using the center of a fuzzy regression model. The proposed model fits intuitive understanding because it makes the model center and the system center coincide.

As mentioned above, our second type model of fuzzy robust regression is built by maximizing possibility grades summation which derived from estimates of the model and data. In other words, the model is built in order to illustrate the possibility distribution. Therefore, our second type model can be built by granule data or fuzzy data, the model treat data as granule data or fuzzy data.

The possibility grade $\mu(y_i, \mathbf{x}_i)$ of $(y_i, \mathbf{x}_i)$ is defined using the center Yi^C and the width W_i of the model.

$$\mu(y_i, \mathbf{x}_i) = 1 - \frac{\left|Y_i^C - y_i\right|}{W_i}$$

Let us calculate the total sum, Z_1, of possibility grades of the fuzzy regression model as follows:

$$Z_1 = \sum_{i=1}^{n} \mu(y_i, \mathbf{x}_i) = \sum_{i=1}^{n} \left(1 - \frac{\left|Y_i^C - y_i\right|}{W_i}\right). \tag{14}$$

In this part, the model is built by maximizing Z_1 defined in Eq. (14), which is the total sum of possibility grades in the objective function of a fuzzy regression model [6, 7]. It is explicit from Eq. (14) that the width, W_i, of the model gets larger if one maximizes Z_1. Therefore, the following function is defined to minimize the sum, Z_2, of the vagueness values W_i of the model:

$$Z_2 = \sum_{i=1}^{n} W_i.$$

Thus, the model is reduced to a bi-objective linear programming problem employing these Z_1 and Z_2 as a multi-objective function.

But, it is easy to build a method to solve the problem using the following weighted sum of the two objective functions as in Eq. (15):

$$\left.\begin{array}{ll} \text{maximize} & Z_3 = \alpha Z_1 - (1-\alpha) Z_2 \\ \text{subject to} & Y_i^L \le y_i \le Y_i^U (i = 1, 2, \ldots, n) \end{array}\right\}. \tag{15}$$

where α is a weight parameter, $0 \le \alpha \le 1$. It is possible to control the shape of the model by changing parameter α of the objective function. Therefore, the value of parameter α is selected heuristically and empirically by a decision maker.

4.2 Analysis of Japanese Main Rivers

A river is created by an erosion effect of precipitation, flowing water, and so on. In addition, topographical factors such as a diastrophism, a tectonic activity, and a fault also build up a river. For this reason, a scale of river is illustrated by a basin area and a drift distance. It is easy to understand that a river with a short drift distance has a wider width and a deeper depth in case of being swollen with a big volume of water. Also, a river may be narrower and shallower in case of a low velocity and a low volume of the water flow.

Let us analyze the basin areas Y and the drift distances X of major rivers in Japan [25] by a regression model.

At beginning, a statistical regression model Y^{LS} based on a least squares method is confirmed as following:

$$Y^{LS} = -0.12 + 1.61X.$$

Here, the basin areas Y and the drift distances X are used a value obtained by logarithmic transformation and these takes a large value. Therefore, it is not problem that the constant term has a negative value.

We employ three models which are the fuzzy regression model proposed by Tanaka and Watada [17] and Tanaka and Guo [19], the fuzzy regression model paying a special attention to the center of the model proposed by Tajima [15], and our second type model of fuzzy robust regression. Therefore, the model is defined as follows:

$$\mathbf{Y} = \left(Y^C, Y^L, Y^U\right) = (\mathbf{a}_0, \mathbf{c}_0, \mathbf{d}_0) + (\mathbf{a}_1, \mathbf{c}_1, \mathbf{d}_1)X,$$

Let us denote fuzzy regression models proposed by Tanaka et al., Tajima, and us by Y^{CM}, Y^{CF}, and Y^{GR}, respectively. These are written as the following three models:

$$\begin{aligned} Y^{CM} &= (1.58, 0, 0) + (1.27, 0.17, 0.46)X, \\ Y^{CF} &= (-0.42, 0.2, 2.52) + (1.60, 0.15, 0)X, \\ Y^{GR} &= (-0.42, 0, 0) + (1.60, 0.15, 0.58)X, \end{aligned}$$

where X and Y denote the drift distance and the basin area, respectively, and the parameter takes the value $\alpha = 0.05$ for the proposed model. Figures 3 and 4, 5 show Y^{CM}, Y^{CF}, and Y^{GR}, respectively. The shape of the models depends on the objective function. In this paper, the fuzziness of the Tanaka's model and our model is the sum of fuzzy coefficients $\mathbf{c} + \mathbf{d}$, which shows the width of the model. In addition this, under the influence of the Yodogawa, the statistical model has a little larger constant term, the regression model is slightly unnatural.

Fig. 3 Y^{CM} on a scale of rivers in Japan

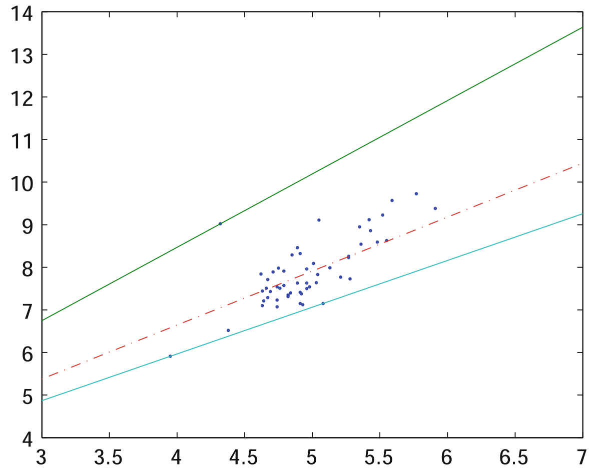


Fig. 4 Y^{CF} on a scale of rivers in Japan

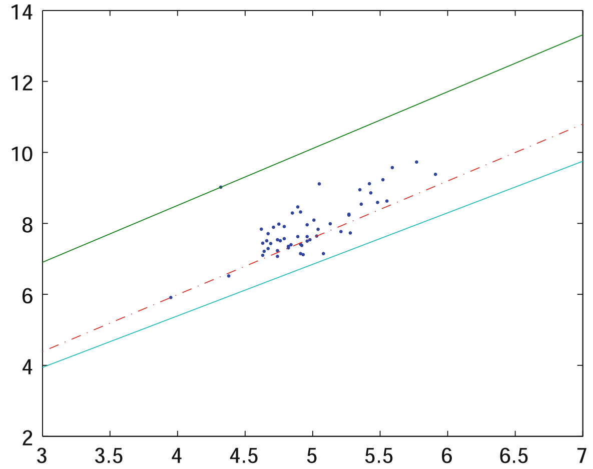


Fig. 5 Y^{GR} on a scale of rivers in Japan

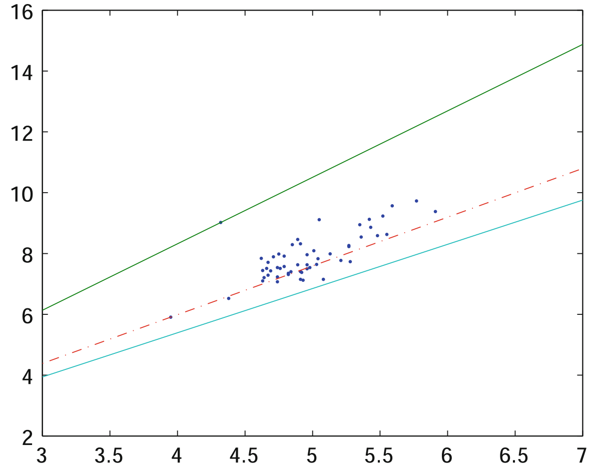


Table 3 Feature of each model

	Y^{CM}	Y^{CF}	Y^{GR}
Sum of all grades	36.904	42.704	43.755
Fuzziness	0.626	2.667	0.731
Residual error	–	344.407	–

The model Y^{CF} can be formulated as follows:

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^{n}\left\{(\mathbf{ax}_i - \mathbf{c}|\mathbf{x}_i| - y_i)^2 + (\mathbf{ax}_i + \mathbf{d}|\mathbf{x}_i| - y_i)^2\right\} \\ &\text{maximize} && \sum_{i=i}^{n}\mu(y_i, \mathbf{x}_i) \\ &\text{subject to} && Y_i^L \le y_i \le Y_i^U (i = 1, 2, \ldots, n). \end{aligned}$$

Table 3 shows the characteristics of these models. The width of vagueness in Table 3 is the summation of fuzzy coefficient $\mathbf{c} + \mathbf{d}$. In the Tajima's model Y^{CF}, the objective function is defined by using the squared distance between the model and the observed value. This value is also shown in Table 3. These values show the characteristics of these models.

Let us discuss the rivers in Japan. The Yodogawa in Osaka is different from other rivers in Japan in the relation between the basin area Y and the drift distance X. The Yodogawa has a short drift distance X but it has a large basin area Y. Therefore, the Yodogawa is placed far from the group of other rivers. Lake Biwa is the water source of the Yodogawa, the Yodogawa carry water to Osaka Bay. The distance between Osaka Bay and Lake Biwa is short, relatively large tributaries are joined with the Yodogawa. Therefore, the three model shapes look distorted.

The conventional fuzzy regression model Y^{CM} describes the system's possibility by minimizing the fuzziness included in the model (Fig. 3). Nevertheless, the center of Y^{CM} is different from the center of possibility distribution because it is influenced by the Yodogawa, and the slope of the regression is small. As Table 3 illustrates, the fuzziness of the model is the lowest among the three models. The proposed model Y^{GR} shows the second lowest fuzziness behind Y^{CM}.

Let us study the central position of the model. The Tajima's model and the proposed model maximizing the total summation of possibility grades show the main trend of the system, even though their graph's slope is somewhat small.

As the Tajima's model can minimize the distance from the samples to the upper and lower boundaries of the model, the right side of the graph shows narrower width because the number of samples is larger. Therefore, the model shows an unnatural possibility.

The proposed model has a large width of possibility because of the influence of the Yodogawa. But Table 3 shows that among the three models, the proposed model naturally describes the system's possibility and the graph's slope. Although this model is built by maximizing the sum of the possibility grade from the model and data, the inclusion constraints might not fit the possibility distribution and the model.

When the inclusion constraints related to the Yodogawa are removed as the model includes all the samples, the following model is obtained:

$$Y^{GR'} = (-0.98, 0, 0) + (1.73, 0.18, 0.44)X.$$

Figure 6 shows this model. The sum of the possibility grades is 42.196. The fuzziness is 0.622. By means of removing the constraints on the proposed model, the sum of the possibility grades becomes smaller. This occurs because the width of the model becomes narrower. The model's construction for mitigating the influence of outlier samples is discussed below.

4.3 Model Removed Influences of Outliers

In the previous section, we illustrated removing the inclusion constraints in order to alleviate the influence of outlier samples. The center of the model is lower than the center of the data distribution in Fig. 6. The outlier, the Yodogawa, is located above the center. In this case, the outlier samples force the center of possibility distribution to move to the opposite side.

Let us discuss this phenomenon using a simple example as shown in Fig. 7. When we construct the membership function using samples including an outlier $\odot$, Fig. 7a is transformed into Fig. 7b.

The center of the model moves to the right in the majority group of samples, that is, the opposite from the outlier sample. This makes the sum of membership grades larger than the initial state.

When we intend to build a model so that the total sum of the possibility grades is maximized, the formation of the model is distorted. As a fuzzy regression model defined so as to include all samples in the model and minimize the total vagueness of the model, then the total sum of possibility grades becomes larger as the width gets wider. The reason is because the form of the membership function is defined so as to set $\mu(a) = 0$ for the outlier sample $\odot$.

In order to mitigate the distortion of the membership function at the outlier sample $\odot$, the formulation of the model is changed as is shown in Fig. 7c.

When the membership function is generated, it is possible to alleviate the influence of outlier samples by a parallel shifting of the graph of the obtained membership function by the step $\mu'(a) = -\beta$ after solving the membership function which has the maximum membership value $1 + \beta$. In other words, when considering the possibility influenced by the outlier sample $\odot$ to build the membership function, the values [0, 1] are used in the membership function. That is, the outlier sample $\odot$ is placed outside of the possibility distribution because of the nature of outlier samples.

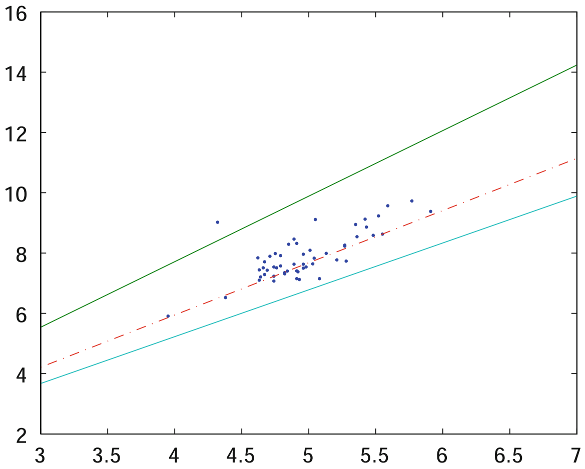


Fig. 6 $Y^{GR'}$ on a scale of rivers in Japan

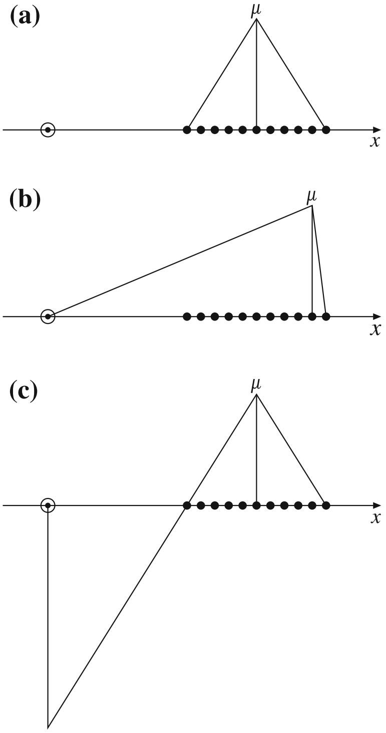


Fig. 7 Outlier mixed in data. **a** Outlier ⊙ is not included in data. **b** Outlier ⊙ is included in data. **c** Removed influences of outlier ⊙

The slope of the straight line in the previous section was smaller than the possibility of the system. Let us consider Fig. 8. The ellipsoid in Fig. 8 denotes the data distribution. When outlier samples are included at left side of the upper

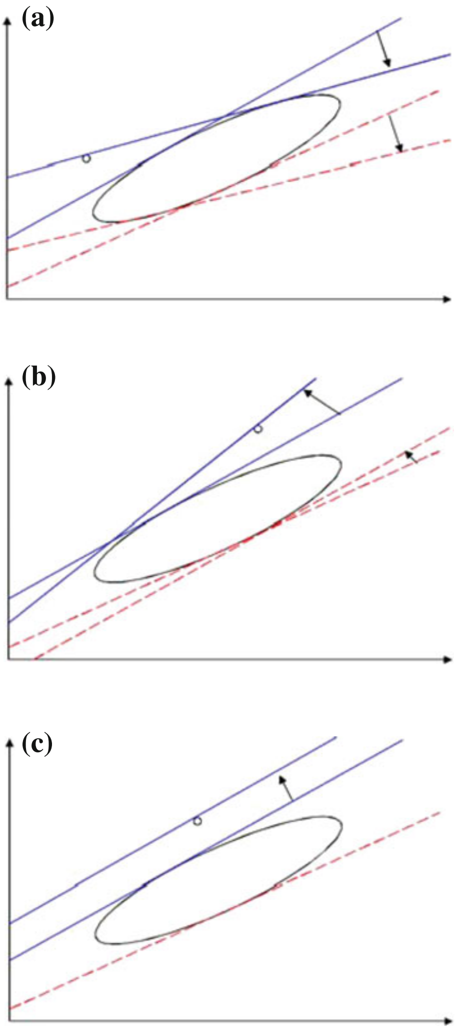


Fig. 8 Location of outlier. **a** Outlier is located in the *upper left portion* of the data distribution. **b** Outlier is located in the *upper right portion* of the data distribution. **c** Outlier is directly above the data distribution

ellipsoid, the fuzzy regression model rotates clockwise and the slope becomes smaller as shown by the arrow in Fig. 8a.

On the other hand, when outlier samples are included at the right side of the upper ellipsoid, the fuzzy regression model rotates counterclockwise and the slope becomes larger, as shown by the arrow in Fig. 8b.

Based upon the above discussion of the fuzzy regression model leads to the conclusion that influence of outlier samples on the maximum of the total sum of possibility grades can be summarized as follows:

- As the grade becomes larger, the width of the model becomes wider.
- The center of the model moves to the opposite of the outlier sample's position.

- The slope of the model's graph is influenced by outlier samples.

When the fuzzy regression model is built by maximizing the possibility grade, its objective is to describe the possibility of the latent system. That means that the emphasis is on the description of the system possibility rather than the inclusion of all the samples in the model. Therefore, it will be allowed to remove the inclusion constraints between sample y and model Y as follows:

$$Y^L \leq y \leq Y^U.$$

When the model is constructed, outlier samples greatly influence fuzzy coefficients as shown in Fig. 8c. Heuristically, we can employ the following objective function under the setting $|\mu(a)| = \beta$.

$$\text{maximize}\, Z_3 = \alpha \sum_{i=1}^{n} |\mu(y_i, \mathbf{x}_i) - \beta| - (1-\alpha)\gamma \sum_{i=1}^{n} W_i.$$

where α is the parameter to decide the selection between the maximization of possibility grade and the minimization of the vagueness of the model. γ is the parameter to tune the difference between the total sum of possibility grades and the vagueness of the model.

Let us consider an example to illustrate the proposed model. Set $\alpha = 0.1$, $\gamma = 2$ and compare cases $\beta = 0.3$ and $\beta = 0.7$. Fuzzy regression models $Y^{GR}_{\beta=0.3}$ and $Y^{GR}_{\beta=0.7}$ are obtained as follows:

$$Y^{GR}_{\beta=0.3} = (-0.98, 0, 0.0) + (1.73, 0.16, 0.45)X,$$
$$Y^{GR}_{\beta=0.7} = (-1.38, 0, 0.0) + (1.80, 0.07, 0.36)X.$$

Parameter α used in the model decides which portion is emphasized between the maximization of the total sum of the possibility grades and the minimization of the total vagueness of the model. When α is set to larger value, the vagueness of the obtained model becomes larger. In the numerical example, the setting of $\alpha = 0.1$ and $\gamma = 2$ makes an obtained model well-balanced.

As β corresponds to $\alpha-$cut, setting β to a larger value makes the width of the model smaller. Figures 9 and 10 depict $Y^{GR}_{\beta=0.3}$ and $Y^{GR}_{\beta=0.7}$, respectively. Parameter β adjusts the width of the model (Table 4). $\beta = 0.3$ shows that the model has too large vagueness. On the other hand, $\beta = 0.7$ shows that non-outlier rivers are placed outside of the possibility distribution. Then, we employed $\beta = 0.5$ to obtain the following model:

$$Y^{GR}_{\beta=0.5} = (-0.98, 0, 0.00) + (1.73, 0.15, 0.39)X.$$

Even this model still shows some influence of the Yodogawa, and the possibility distribution of samples is appropriately expressed (Fig. 11).

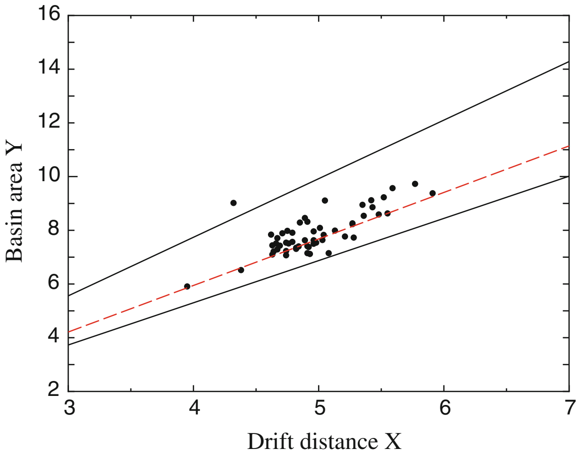


Fig. 9 Improved model 1 ($\beta = 0.3$)

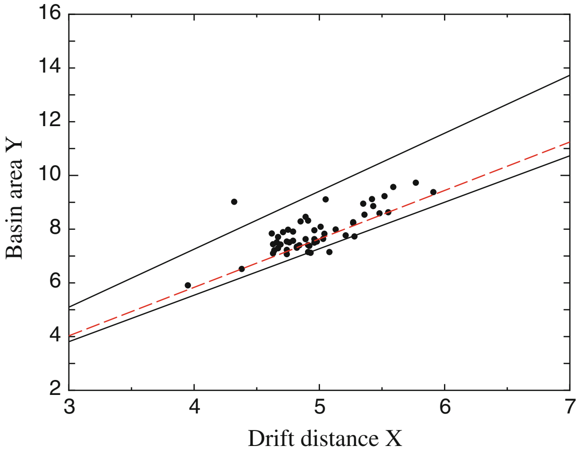
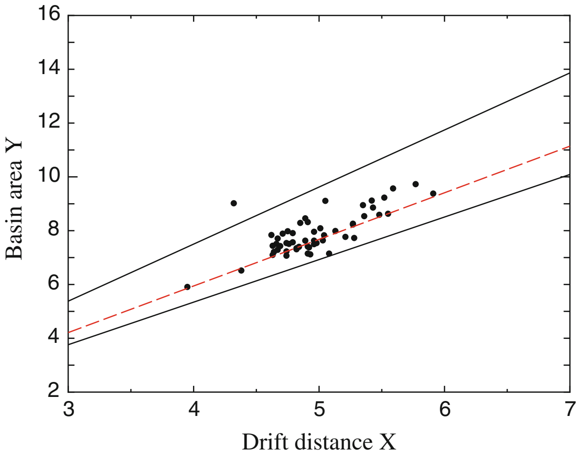


Fig. 10 Improved model 2 ($\beta = 0.7$)

Table 4 Features of improved model

	Improved model	
	1	2
β	0.3	0.7
Sum of all grades	42.531	38.960
Value of Obj. Func.	2.826	2.585

Fig. 11 Optimum model ($\alpha = 0.1$, $\beta = 0.5$)



5 Conclusions

In this chapter, we proposed two type models of fuzzy robust regression. The first type model of fuzzy robust regression treats with granule data or fuzzy data after removal of ill effects of extraordinary data by genetic algorithm. The distance concept is an easy-to-use and feasible way for data including a vagueness, this was confirmed by the first type model of fuzzy robust regression.

The relation between Asian economy and environment was analyzed by the first type model. Relation between Asian economy and environment was explained by the first type model. In this analysis, employing the distance between a conventional fuzzy regression model and slightly different character countries, the relation between Asian economy and environment was explained by the first type model.

The second type model of fuzzy robust regression illustrates the possibility of the target system by its triangular membership function. This second model is built in order to get the maximum degree of coincidence between a second type model and a possibility distribution. Therefore, this second type model is not able to handle only real-valued data but also granule data or fuzzy data.

The features of Japanese major river was analyzed by the second type model, the basin area and the drift distance was used. The center, the upper limit and the lower limit of the system was reveals by the second type model. And the feature of Japanese major river was explained. Since Yodogawa has short drift distances relative to large basin area, the possibility limits was spread. However, the problem was solved by the approach described above. This, the feature of Japanese major river became clear.

Finally, we can conclude our fuzzy robust regression models are able to describe a target possibility by granule data or fuzzy data.

References

1. Coppi, R., D'Urso, P., Giordani, P., Santoro, A.: Least squares estimation of a linear regression model with LR fuzzy response. Comput. Stat. Data Anal. **51**(1), 267–286 (2006)
2. Diamond, P.: Least squares and maximum likelihood regression for fuzzy linear models. In: Kacprzyk, J., Fedrizzi, M. (eds.) Fuzzy Regression Analysis. Omnitech Press, Warsaw, pp. 137–151 (1992)
3. Diamond, P.: Fuzzy least squares. Inf. Sci. **46**(3), 141–157 (1988)
4. D'Urso, P., Gastaldi, T.: A least-squares approach to fuzzy linear regression analysis. Comput. Stat. Data Anal. **34**(4), 427–440 (2000)
5. Guo, P., Tanaka, H.: Fuzzy DEA: a perceptual evaluation method. Fuzzy Sets Syst. **119**(1), 149–160 (2001)
6. Hasuike, T., Katagiri, H., Ishii, H.: Multiobjective random fuzzy linear programming problems based on the possibility maximization model. J.Adv. Comput. Intell. Intell. Inf. **13**(4), 373–379 (2009)
7. Honda, A., Okazaki, Y.: Identification of fuzzy measures with distorted probability measures. J.Adv. Comput. Intell. Intell. Inf. **9**(5), 467–476 (2005)
8. Hong, D.H., Hwang, C., Ahn, C.: Ridge estimation for regression models with crisp inputs and Gaussian fuzzy output. Fuzzy Sets Syst. **142**(2), 307–319 (2004)
9. Inuiguchi, M., Tanino, T.: Interval linear regression methods based on minkowski difference: a bridge between traditional and interval linear regression models. Kybernetika **42**(4), 423–440 (2006)
10. Ishibuchi, H., Tanaka, H.: Several formulations of interval regression analysis. In: Proceedings of Sino–Japan Joint Meeting on Fuzzy Sets and Systems, Section B2-2 (1990)
11. Imoto, S., Yabuuchi, Y., Watada, J.: Fuzzy regression model of R&D project evaluation. Appl. Soft Comput. J. **8**(3), 1266–1273, 2008 (Special Issue on Forging New Frontiers)
12. Modarres, M., Nasrabadi, E., Nasrabadi, M.M.: Fuzzy linear regression model with least square errors. Appl. Math. Comput. **163**(2), 977–989 (2005)
13. Science and Technology Agency: Energy Consumption and the Global Environment in Asia (1992) (in Japanese)
14. Shibata, K., Watada, J., Yabuuchi, Y.: A fuzzy robust regression approach for evaluating electric and electronics corporations. J. Jpn. Ind. Manag. Assoc. **59**(1), 58–67 (2008)
15. Tajima, H.: A proposal of fuzzy regression model. In: Proceedings of the Vietnam–Japan Bilateral Symposium Fuzzy Systems and Applications, pp. 383–389 (1998)
16. Tanaka, H., Watada, J., Asai, K.: Linear Regression Analysis by Possibilistic Models. In: Kacprzyk, J., Orlovsky, S. (eds.) pp. 186–199 (1987)
17. Tanaka, H., Watada, J.: Possibilistic linear systems and their application to the linear regression model. Fuzzy Sets Syst. **27**(3), 275–289 (1988)
18. Tanaka, H., Ishibuchi, H., Yoshikawa, S.: Exponential possibility regression analysis. Fuzzy Sets Syst. **69**(3), 305–318 (1995)
19. Tanaka, H., Guo, P.: Possibilistic Data Analysis for Operations Research. Phisica-Verlag, Heidelberg (1999)
20. Toyoura, Y., Watada, J., Yabuuchi, Y., Ikegame, H., Sato, S., Watanabe, K., Tohyama, M.: Fuzzy regression analysis of software bug structure. CEJOR **12**(1), 13–23 (2004)
21. Toyoura, Y., Watada, J., Khalid, M., Yusof, R.: Formulation of linguistic regression model based on natural words. Soft. Comput. **8**(10), 681–688 (2004)
22. Watada, J.: Fuzzy time-series analysis and forecasting of sales volume. In: Kacprzyk, J., Fedrizzi, M. (eds.) Fuzzy Regression Analysis, pp. 211–227 (1992)
23. Watada, J., Yabuuchi, Y.: fuzzy robust regression analysis based on a hyperelliptic function. In: The Proceeding of Third IEEE International Conference on Fuzzy Systems, pp. 1841–1848 (1995)
24. Watada, J., Pedrycz, W.: Handbook of Granular Computing. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) pp. 719–740 (2008)

25. Water Handbook Editorial Committee.: Water Handbook, p. 95. Maruzen, Tokyo (2003) (in Japanese)
26. Yabuuchi, Y., Watada, J.: Fuzzy robust regression analysis based on a hyperelliptic function. In: Proceedings of the 4th IEEE International Conference on Fuzzy Systems, pp. 1841–1848 (1995)
27. Yabuuchi, Y., Watada, J.: Possibility robust regression analysis. In: Proceedings of the 20th International Conference on Computers and Industrial Engineering, pp. 9–12 (1997)
28. Yabuuchi, Y., Watada, J.: Model building based on central position for a fuzzy regression model. Proc. Czech–Jpn. Semin. **2006**, 114–119 (2006)
29. Yabuuchi, Y., Watada, J.: Fuzzy regression model building through possibility maximization and its application. Innovative Comput. Inf. Control Express Lett. **4**(2), 505–510 (2010)
30. Yabuuchi, Y., Watada, J.: Fuzzy robust regression model by possibility maximization. J.Adv. Comput. Intell. Intell. Inf. **15**(4), 479–484 (2011)
31. Yabuuchi, Y., Watada, J.: Japanese economic analysis by possibilistic regression model which built through possibility maximization. J.Adv. Comput. Intell. Intell. Inf. **16**(5), 576–580 (2012)

Part II
Architectures

The Role of Cloud Computing Architecture in Big Data

Mehdi Bahrami and Mukesh Singhal

Abstract In this data-driven society, we are collecting a massive amount of data from people, actions, sensors, algorithms and the web; handling "Big Data" has become a major challenge. A question still exists regarding when data may be called big data. How large is big data? What is the correlation between big data and business intelligence? What is the optimal solution for storing, editing, retrieving, analyzing, maintaining, and recovering big data? How can cloud computing help in handling big data issues? What is the role of a cloud architecture in handling big data? How important is big data in business intelligence? This chapter attempts to answer these questions. First, we review a definition of big data. Second, we describe the important challenges of storing, analyzing, maintaining, recovering and retrieving a big data. Third, we address the role of Cloud Computing Architecture as a solution for these important issues that deal with big data. We also discuss the definition and major features of cloud computing systems. Then we explain how cloud computing can provide a solution for big data with cloud services and open-source cloud software tools for handling big data issues. Finally, we explain the role of cloud architecture in big data, the role of major cloud service layers in big data, and the role of cloud computing systems in handling big data in business intelligence models.

Keywords Big data · Cloud computing · Cloud architecture · Business intelligence

M. Bahrami (✉) · M. Singhal
Cloud Lab, Electrical Engineering and Computer Science Department,
University of California, Merced, CA, USA
e-mail: mbahrami@ucmerced.edu

M. Singhal
e-mail: msinghal@ucmerced.edu

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_13,

1 Introduction

Capturing data from different sources allows a business to use Business Intelligence (BI) [1] capabilities. These sources could be consumer information, service information, products, advertising logs, and related information such as the history of product sales or customer transactions. When an organization uses BI technology to improve services, we characterize it as a "smart organization" [1]. The smart features of these organizations have different levels which depend on the accuracy of decisions; greater accuracy of data analysis provides "smarter" organizations.

For this reason, we are collecting a massive amount of data from people, actions, sensors, algorithms, and the web which forms "Big Data." This digital data collection grows exponentially each year. According to [2], big data refers to datasets whose size is beyond the ability of typical database software tools and applications to capture, store, manage and analyze.

An important task of any organization is to analyze data. Analysis could change a large volume of data to a smaller amount of valuable data, but we still require collecting a massive amount of data.

Big data has become a complex issue in all disciplines of science. In scientific big data, several solutions have been proposed to overcoming big data issues in the field of life sciences [3, 4], education systems [5], material sciences [6], social networks [7, 8] and.

Some examples of the significance of big data for generating, collecting and computing are listed as follows:

Big data generation and collection:

- It is predicated that data production will be 44 times greater in 2020 than it was in 2009 [9]. This data could be collected from variety resources, such as traditional databases, videos, images, binary files (applications) and text files;
- It is estimated 235 TB of data were collected by the U.S. Library of Congress in April 2011 [10];
- Facebook stores, accesses, and analyzes 30+ Petabytes of user generated data [11] which includes a variety of data, such as images, videos and texts.

Computing on big data:

- In 2008, Google was processing 20,000 TB of data (20 PB) per day [12].
- Decoding the human genome originally took 10 years to process; now it can be achieved in 1 week [13] with distributing computing on big data.
- IDC[1] estimates that by 2020, business-to-business and business-to-consumer transactions on the Internet will reach 450 billion per day [14].

[1] International Data Corporation (IDC) is an American market research, analysis and advisory firm specializing in information technology, telecommunications, and consumer technology.

- Big data is a top business priority and drives enormous opportunities for business improvement. Wikibon's own study projects that big data will be a $50 billion business by 2017 [15].
- Macy's Inc. provides a real-time pricing. The retailer adjusts pricing in near real-time for 73 million items for sale based on demand and inventory [16].
- The major VISA process more than 172,800,000 card transactions each day [17].

The most public resource data are available on the Internet, such as multimedia steam data, social media data and text. This variety of data shows we are not facing only structured data, but also unstructured data, such as multimedia files (including video, audio and images), and Twitter and Facebook comments. Unstructured data causes complexity and difficulty in analyzing big data. For example, a corporation analyzes user comments and user shared data on social media that could recognize customer favorites and provide best offers.

To collect and process big data, we can use Cloud Computing Technology. Cloud computing is a new paradigm for hosting clusters of data and delivering different services over a network or the Internet. Hosting clusters of data allows customers to store and compute a massive amount of data on the cloud. This paradigm allows customers to pay on pay-per-use basis and enables them to grow (or shrink) their computing and storage needs on demand. These features allow customers to pay the infrastructure for storing and computing based on their current capacity of big data and transactions.

Currently, capturing and processing big data are related to improving the global economy, science, social affair, education and national security; processing of big data allows us to propose accurate decisions and acquire knowledge from raw data.

This chapter aims to show the role of cloud computing in dealing with big data and intelligent computing. This chapter is organized as follows: Sect. 2 discusses a definition and characteristics of big data. In Sect. 3, we discuss important opportunities and challenges in handling big data. In Sect. 4, we discuss cloud computing and key architectural components for dealing with big data. In this section, we review how each service layer of a cloud computing system could handle big data issues. Also, we provide a list of services and tools for dealing with big data. Finally, in Sect. 5, we review some major cloud computing issues.

2 Big Data Definition

Often big data is characterized by "4 V's" [18] which stand for:

- "**Volume**" which indicates a very large volume of data;
- "**Velocity**" which indicates the speed for data processing in terms of response time. This response time could be a batch, real-time or stream response-time;
- "**Variety**" which indicates heterogeneity in data that we have collected for processing and analysis this data variety includes structured, unstructured and semi-structured data;

- "**Veracity**" which indicates level of accuracy in the data. For example, a sensor that generates data can have a wrong value rather than provides an accurate data.

Big data could have one or multiple of the above characteristics. For example, storing and computing on social data could have a very large volume of data (*volume*) and specific response-time for computing (*velocity*) but it may not have *variety* and *veracity* characteristics.

Another example, analyzing public social media data regarding the purchase history of a customer could provide a future favorite purchase list when she searches for a new product. In this case, big data have all characteristics: *volume of data*, because collecting a massive amount of data from public social media networks; *velocity*, because response-time limited to near real-time when a customer search a product; *variety*, because big data may come from different sources (social media and purchase history); *lack of veracity*, because data from customers in social media networks may have uncertainty. For instance, a customer could like a product in a social media network, not because this is the product of her choice, but because of this product is used by her friend.

Another important question in big data is, "*How large is big data?*" We can answer this question based on our current technology. For example, Jacobs [19] states in the late 1980s at Columbia University that they stored 100 GB of data as big data via an IBM 3850 MSS (Mass Storage System), which costs $40 K per GB. In 2010, the Large Hadron Collider (LHC) facility at CERN produced 13 PB of data [20]. So what we call big data depends on the cost, speed and capacity of existing computing and storage technologies. For example, in the 1980s, 100 GB was big data because the storage technology was expensive at that time and it had low performance. However, by 2010, the LHC processed 13 PB as a big data which has 1.363×10^5 times more volume than IBM 3850 MSS big data in 1980s.

At this time, we can refer to 13 PB at CERN. In addition, we can also refer to a text file with 10 GB size as big data because a regular text editor could not handle this file size. So the definition of big data is not only *a massive amount of data* but also depends on *what the technology and which size of big data that technology could handle*.

3 Big Data Opportunities and Challenges

On one hand, when we collect big data, we have an opportunity to make an accurate decision through BI. BI is a set of theories and technologies that aim to transfer data from raw-data into meaningful and useful information for business processes (BP). BI became popular in the 1990s, and Business Analytics (BA), which is an analytical component in BI, became popular in the 2000s. In the traditional model, the queries are pre-defined to confirm or refuse a query's hypotheses, but Online Analytical Processing (OLAP) analysis emerges as an approach to answer complex analytical queries. For example, in a car accident we can make a decision about the incident based on driver information. However, when we collect GPS information, engine information and driver information, we can make a more accurate decision about an accident. Also, if we collect more information, we can trust our decision more (*veracity*). In a second example, Volvo provided performance and fault monitoring for predictive warranty analysis [22]. In another example, sensor data from a cross-country flight (New York to Los Angeles) generate 2.499 billion Terabyte per year [23] (*volume*) from different sensors (*variety*), which could be provided from reliable sensors (*veracity*) or unreliable sensors (*lack of veracity*). Often the processing of this data is real-time (*velocity*) and this computing could be processed by an aircraft's server or by a ground's servers.

Collection of information cannot only help us to avoid car accidents, but also could help to make an accurate decision in any systems, such as business financial systems [15], education systems [24], and treatment systems, e.g., Clinical Decision Support Systems [25].

Some important opportunities are provided by big data. They are listed as follows:

- Analyze big data to improve business processes and business plans, and to achieve business plan goals for a target organization (The target organization could be a corporation, industry, education system, financial system, government system or global system.)
- Reduce bulk data to a valuable smaller amount of data
- Provide more accurate decisions by analyzing big data
- Prevent future system failures by predicting big data

On the other hand, we have several issues with big data. The challenges of big data happened in various domains including *storing of big data*, *computing on big data* and *transferring of big data*. We discuss these issues below.

3.1 Storage Issues

A database is a structured collection of data. In the late 1960s, flat-file models which were expensive and slow, used for storing data. For these reasons, relational databases emerged in the 1970s. Relational Database Management Systems (RDBMS) employ Structured Query Language (SQL) to store, edit and retrieve data.

Lack of support for unstructured data led to the emergence of new technologies, such as BLOB (Binary Large Object) in the 2000s. Unstructured data may refer to multimedia data. Also unstructured data may refer to irregularly or randomly repeated column patterns that vary from row to row within each file or document. BLOB could store all data types in most RDBMS.

In addition, a massive amount of data could not use SQL databases because retrieving data and analyzing data takes more time for processing. So "NOSQL", which stands for "Not Only SQL" and "Not Relational", was designed to overcome this issue. NOSQL is a scalable partitioned table that could distribute data over many servers. NOSQL is implemented for cloud computing because in the cloud, a data storage server could be added or removed anytime. This capability allows for the addition of unlimited data storage servers to the cloud.

This technology allows organizations to collect a variety of data but still increasing the volume of data increases cost investment. For this reason, capturing high-quality data that could be more useful for an organization rather than collecting a bulk of data.

3.2 Computing Issues

When we store big data, we need to retrieve, analyze and modify it. The important part of collecting data is analyzing big data and converting raw data into valuable information that could improve a business process or decision making. This challenge can be addressed by employing a cluster of CPUs and RAMs in cloud computing technology.

High-Performance Computing (HPC) is another technology that provides a distributed solutions by different computing models, such as traditional (e.g. Grid Computing) or cloud computing for scientific and engineering problems. Most of these problems could not process data in a polynomial time-complexity.

3.3 Transfer Issues

Transfer of big data is another issue. In this challenge, we are faced with several sub-issues: Transfer Speed, which indicates how fast we can transfer data from one location/site to another location/site. For example, transferring of DNA, which is a type of big data, from China to the United States has some delay in the backbone of the Internet, which causes a problem when they receive data in the United States [26]. BGI (one of the largest producers of genomic data, Beijing Genomics Institute in Shenzen, China) could transfer 50 DNAs with an average size of 0.4 TB through the Internet in 20 days, which is not an acceptable performance [26].

Traffic Jam: transfer of big data could happened between two local sites, cities or worldwide via the Internet but between any locations this transfer will result in a very large traffic jam.

Accuracy and Privacy: Often we transfer big data through unsecured networks, such as the Internet. Data transfers through the Internet must be kept secure from unauthorized access. Accuracy aims to transfer data without missing any bits.

4 Dealing with Big Data

Several *traditional* solutions have emerged for dealing with big data such as Supercomputing, Distributed Computing, Parallel Computing, and Grid Computing. However, elastic scalability is important in big data which could be supported by cloud computing services which are described in Sect. 4.2. Cloud computing has several capabilities for supporting big data which are related to handling of big data. Cloud computing could support two major issues of big data, which are described in Sect. 3 including storing of big data and computing of big data. Cloud computing provides a cluster of resources (storage and computing) that could be added anytime. These features allow cloud computing to become an emerging technology for dealing with big data.

In this section, we will first review important features of cloud computing systems and a correlation of each of them to big data. Second, we discuss a cloud architecture and the role of each service layer in handling big data. Finally, we review implementation models of cloud computing systems as they relate to handling big data.

4.1 Cloud Computing System Features

The major characteristics of cloud computing as defined by the U.S. National Institute of Standards and Technology (NIST) [27] are as follows:

4.1.1 On-demand Elastic Service

This characteristics show the following features: (i) an economical model of cloud computing which enables consumers to order required services (computing machines and/or storage devices). The service requested service could scale rapidly upward or downward on demand; (ii) it is a machine responsibility that does not require any human to control the requested services. The cloud architecture manages on-demand requests (increase or decrease in service requests), availability, allocation, subscription and the customer's bill.

This feature is interesting for a start-up businesses, because this feature of cloud computing systems allows a business to start with traditional data or normal datasets (in particular start-up business) and increase their datasets to big data as they receive requests from customers or their data grows during the business progress.

4.1.2 Resource Pooling

A cloud vendor provides a pool of resources (e.g., computing machines, storage devices and network) to customers. The cloud architecture manages all available resources via global and local managers for different sites and local sites, respectively.

This feature allows big data to be distributed on different servers which is not possible by traditional models, such as supercomputing systems.

4.1.3 Service Accessibility

A cloud vendor provides all services through broadband networks (often via the Internet). The offered services are available via web-based model or heterogeneous client applications [28]. The web-based model could be an Application Programming Interface (API), web-services, such as Web Service Description Language (WSDL). Also heterogeneous client applications are provided by the vendors. Customers could run applications on heterogeneous client systems, such as Windows, Android and Linux. This feature enables partners to contribute to big data. These partners could provide cloud software applications, infrastructure or data. For example, several applications from different sites could connect to a single-data or transparent multiple-data warehouse for capturing, analyzing or processing of big data.

4.1.4 Measured Service

Cloud vendors charge customers by a metering capability that provides billing for a subscriber, based on pay-per-use model. This service of cloud architecture manages all cloud service pricing, subscriptions and metering of used services. This capability of cloud computing system allows an organization to pay for the current size of datasets and then pay more when dataset size increases. This service allows customers to start with a low investment.

4.2 Cloud Architecture

Cloud computing technology could provide by a vendor that enables IT departments to focus on their software development rather than hardware maintenance, security maintenance, recovery maintenance, operating systems and software upgrades. Also, if an IT department establishes a cloud computing system in their organization, could help them to handle big data.

The Architecture of a cloud computing system is specific to the overall system and requirements of each component and sub-components. Cloud architecture allows cloud vendors to analyze, design, develop and implement big data.

Cloud vendors provide services through service layers in cloud computing systems. The major categories are divided into four service layers: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS)

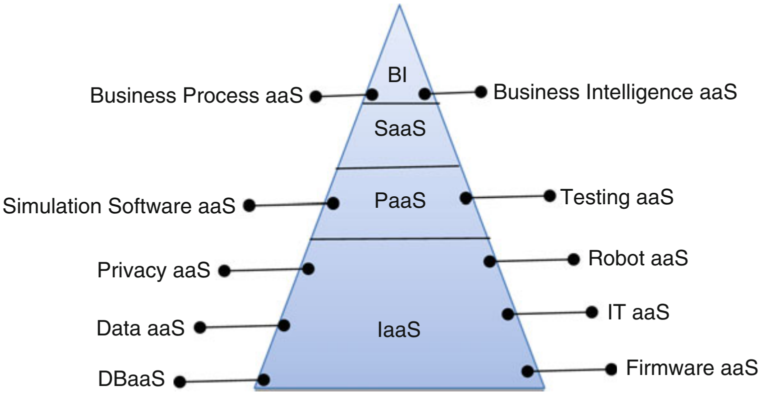


Fig. 1 Cloud services

and Business Intelligence (BI) and other service layers assigned to the major service layers as shown in Fig. 1, such as Data-as-a-Service(DaaS) assigned to IaaS layer. Description of each service discussed in Sect. 4.2.5.

4.2.1 The Role of Infrastructure as a Service

The IaaS model offers storage, processors and fundamental hardware to the cloud customers. This model covers several services, such as firmware, hardware, utilities, data, databases, resources and infrastructure. This model allows clients to install operating systems, receive quoted infrastructure, and develop and deploy required software applications. This model is often implemented via Virtualization, which enables multi users/tenants work on share machines with his own privacy.

The IaaS model provides several opportunities for big data: (1) ***storage data*** this feature allows customers to store big data. Storage on the cloud computing system enables customers to store, retrieve and edit big data by employing a cluster of storage devices. These clusters could be added or removed dynamically; (2) ***hardware*** this feature enables customers have an access to a resource pool of hardware for big data. This feature could be used for capture data, such as through sensors, Radio-Frequency Identifications (RFIDs) or Communication-as-a-Service (CaaS). The CaaS is responsible for the required hardware and software for delivering Voice-over-IP (VoIP), audio and video conferencing. The hardware feature also provides network access and network traffic control that could to transfer big data.

Amazon Elastic Compute Cloud (Amazon EC2) provides virtual and scalable computing systems at the IaaS. Amazon EC2 customers could define instances of a variety of operating systems (OSs). Each OS and required hardware, such as CPUs and RAMs could be customized by a customer on the fly. Customers should create an Amazon Machine Image (AMI) in order to use Amazon EC2. The AMI contains the required applications, operating systems (the customer could select various operating systems such as Windows or Linux versions), libraries, data and system configuration. Amazon EC2 uses Amazon S3, which is a cloud storage service and stores data and uploads AMI into S3.

The impact of big data in this service layer is higher than other service models in cloud computing systems, because IaaS users could access and define the required *data framework*, *computing framework* and *network framework*.

In a *data framework*, users could define structured data, unstructured data and semi-structured data. Structured and semi-structured data could be defined via traditional databases, such as RDBMS and OODBMS. In these models, structured data stored which has a schema before adding data to the databases. All of data frameworks and in particular unstructured data could be defined by cloud databases, such as Hadoop which is based on MapReduce programming model. MapReduce programming language technique allows storing data on a cluster of resources. The implementation model of MapReduce is provided by Hadoop which is provided a category of open-source database, applications and analytics tools.

In *computing framework*, users have full-permission for developing, installing and running new application for computing purposes. Each application could reserve a cluster of CPUs and RAMs. Several tools and databases with analysis tools emerge to provide computing framework on big data. For example, Hive is an open-access "SQL-like" BI tools that allows BI applications to run query on Hadoop data. Other example, Pig is another open-source platform that allows analyzing on big data by a "Perl-language-like" feature.

In *network framework, users have a* significant benefit, because they have access to required network control, such as network cards and the Internet connectivity. For example they could access to regular network transfer infrastructure such as Optical Carrier (OC) 768 backbone [29], which is capable of transferring 39,813.12 MB/s.

This accessibility to data, computing and network framework allows the users to control require hardware like an administrator in IT department. However, these users could handle infrastructure without worrying about maintenance.

4.2.2 The Role of Platform as a Service

PaaS is a platform that provided by cloud vendor. The PaaS model does not require users to setup of any software, programming language, environment application, designer, tools or application. Developers use vendor's platform, library and programming language for developing their applications. This model provides a software application for outgrowth of the cloud applications delivery. PaaS allows developer to focus on software application development, without worrying about operating system maintenance like in IaaS. The PaaS provides services for software programmers to develop and deploy their applications with an abstraction on the hardware layer.

The role of PaaS in handling big data is less than IaaS, because some restrictions and limitations are applied to PaaS users in order to work on the data framework, computing framework and transfer frameworks. In this service layer, users are limited to cloud vendor frameworks. For example, Google App Engine provides a platform which supports Python, Java, PHP, Go and MySQL compatible Cloud SQL to develop applications. So, in this service layer, users could not access other languages, such as C# or C++ and server hardware. However, developers still could build, deploy and run their scalable applications on the cloud computing systems. These applications could capture a massive amount of data from anywhere and use a cluster of CPUs for computing and analytics of big data.

4.2.3 The Role of Software as a Service

The traditional model of software is to purchase software applications and install them on the local computer. However, SaaS model provides applications in the cloud though a network and does not require customers to install applications on their local computers.

According to Microsoft, SaaS model could be divided to the following categories (lower-level to higher-level) [30]:

- *Ad hoc/Custom* which supports by minimum requirement to migrate traditional and client/server application to this level. Ad hoc/Custom models allow developer to build their application based on ad hoc or peer-to-peer technology;
- *Configurability* which provides more flexibility through configuration metadata and supports peer-to-peer technology;
- *Multi-tenancy* which adds multi-tenancy to the configuration level, and a single instance of application allows serving all the vendor's consumers; and
- *Scalability* which supports all other lower-levels. In addition, this level supports scalability through architectural design that adds a capability of dynamic load-balancing for growing or shrinking cloud servers. Most applications in the cloud are developed at this level.

The impact of SaaS is less than PaaS, because in this service layer, users could use provided applications and resources. This service layer is limited to developers. However, users still could work on big data that could be added before or captured by provided infrastructure. For example, Google Apps, such as Gmail, provides services on the web and users could not add or manipulate capturing data from server. Users are limited to web-based interface for email processes such as sending an email.

4.2.4 The Role of Business Intelligence

The BIaaS layer sits on the top of cloud architecture service layers and aims to provide the required analytic models for cloud customers.

Information granularity as Pedrycz defined [31] is a structure which plays a key role in human cognitive and decision-making computing. The BI service layer could provide a platform for information granularity on the cloud computing and in particular granular computing, which is a processing of complex information entities. Unlike the traditional computing, cloud computing by granular computing on big data may provide a significant result. For example, Bessis at al. [32] propose a big picture by collecting big data and using cloud computing for managing disasters.

Cloud computing could provide the following information granularity and granular computing infrastructures [31]:

- A granular description of data and pattern classification by non-SQL databases, such as SciDB [33];
- A representation of information granules by migrating traditional applications to the cloud;
- Different granular architecture and development by collecting information from different sources and computing with high quality rather than traditional models which were working with a limited computing resource;
- Collaborative and linguistic models of decision-making by collecting information from different sources at the cloud storages.

The information-processing level [34], which is encountering a number of conceptual and algorithmic layers indexed by the size of information granular, could be high if a cloud application provides a computing model. However, if a cloud application provides only a storage model, this impact and granular computing will be low. For example, when an application provides a service for collecting data from financial consumers and running an analytical model on this data to make a decision about investment, cost and profit, this application has a high-level BIaaS impact. For instance, Xu et al. [35] present "Big Cloud based Parallel Data miner (BC-PDM)" which is a framework for integrating data mining applications on MapReduce and HDFS (Hadoop File System) platforms.

Cloud based BI could reduce the total development cost, because cloud computing systems provide environment for agile development and reduce the maintenance cost. Also, the BI could not be implemented on a traditional system, because the current volume of data for analysis is massive. BI-as-a-Service [36] is other example that shows how the BI could migrate to the cloud computing systems as a software application in the SaaS layer.

One of the major challenges with traditional computing is analysis of big data. Cloud computing at BIaaS layer could handle this issue by employing a cluster of computing resources. For example, SciDB [33] is an open-source and cloud-based database management system (NOSQL DBMS) for scientific application with several functions for analyzing of big data, such as astronomy, remote sensing and climate modeling.

4.2.5 Other Service Layers

The major service models of cloud computing are BIaaS, IaaS, PaaS and SaaS. As shown in Table 1, we assigned each service to the major service models.

4.2.6 Big Data Tools

The Table 2 shows some big data open-source tools which are provided through cloud computing infrastructures. Most of the tools are provided by Apache[2] and released under the Apache License. We categorized each tool based on those applications of big data.

4.3 Implementation Models of Cloud Computing Systems

A Cloud computing system based on infrastructure location could be implemented as Private, Public or Hybrid cloud.

The ***private model*** is a local implementation of cloud computing system. In this model, hardware is located in local data centers and uses cloud software applications to provide service to local users. This model is the best option for consumers who needs cloud computing capabilities with low-risk in IT departments because this model allows an IT department to migrate from the traditional model to the cloud computing system and does not require data to be migrated to another location (such as cloud vendor location). This model is implemented for local trusted users. This model still allows scalability, on-demanded self-service, and elastic service. However, this model requires high investment in maintenance, recovery, disaster control, security control, and monitoring.

In addition, the private cloud computing model enables an IT department to handle a local organization's big data by its own infrastructure, such as the storage of big data and computing big data. This model provides a flexible resource assignment and could enhance the resource availability.

Several open source applications have been developed for establishing private cloud computing based on IaaS and SaaS service layers. For example, CloudIA is a private cloud computing system at (HFU) [48]. The targeted users of the CloudIA project are HFU staff and students running e-Learning applications, and external people for collaboration purposes.

The ***public model*** is a regular model of cloud computing system. This model is provided by cloud vendor who supports billing and a subscription system for public users. This model, unlike a private model, does not require high investment,

[2] http://apache.org/.

Table 1 Other service layers in cloud architecture

Service name	Related to	Service description and offers	Role of service in big data
Business-process-as-a-service (BPaaS) [37]	BIaaS	Automated tool support	Analysis of big data
Business-intelligence-as-a-service (BIaaS) [38]	BIaaS	Integrated approaches to management support	Analysis of big data
Simulation software-as-a-service (SimSaaS) [39]	SaaS	Simulation service with a MTA configuration model	Analysis of big data
Testing-as-a-service (TaaS) [40]	SaaS	Software testing environments	Test big data tools
Robot-as-a-service (RaaS) [41]	PaaS	Service-oriented robotics computing	Action on big data
Privacy-as-a-service (PaaS) [42]	PaaS	A framework for privacy preserving data sharing with a view of practical application	Big data privacy
IT-as-a-service (ITaaS) [43]	IaaS	Outsource IT department's resource (on grid infrastructure that time)	Maintaining of big data
Hardware-as- a service (HaaS) [44]	IaaS	A transparent integration of remote hardware that is distributed over multiple geographical locations into an operating system	Capturing and maintaining of big data
Database-as-a-service (DBaaS) [45]	IaaS	1. A workload-aware approach to multi-tenancy 2. A graph-based data partitioning algorithm 3. An adjustable security scheme	Storing big data
Data-as-a-service (Daas) [46]	IaaS	Analyzing major concerns for data as a service	Storing big data
Big-data-as-a-service [47]	All layers	Service-generate for big data	Generate big data

Table 2 Big data tools

Big data tools	Description[1]
	Data analysis tools
Ambari[2]	A web-based tool for provisioning, managing, and monitoring apache hadoop clusters
Avro[3]	A data serialization system
Chukwa[4]	A data collection system for managing large distributed systems
Hive[5]	A data warehouse infrastructure that provides data summarization and ad hoc querying
Pig[6]	A high-level data-flow language and execution framework for parallel computation
Spark[7]	A fast and general compute engine for hadoop data. Spark provides a simple and expressive programming model that supports a wide range of applications, including ETL, machine learning, stream processing, and graph computation
ZooKeeper[8]	A high-performance coordination service for distributed applications
Actian[9]	An analytics platform which accelerates the analytics value chain from connecting to massive amounts of raw big data all the way to delivering actionable business value
HPCC[10]	Provide high-performance, data-parallel processing for applications utilizing big data
	Data mining tools
Orange[11]	A data visualization and analysis for novice and experts
Mahout[12]	A scalable machine learning and data mining library
KEEL[13]	An assess-evolutionary algorithm for data mining problems
	Social network tools
Apache kafka	A unified, high-throughput, low-latency platform for handling real-time data feeds
	BI tools
Talend[14]	A data integration, data management, enterprise application integration and big data software tools and services
Jedox[15]	An analyzing, reporting and planning functions
Pentaho[16]	A data integration, business analytics, data visualization and predictive analytics
rasdaman[17]	A multi-dimensional raster data (arrays) of unlimited size through an SQL-style query language
	Search tools
Apache lucene[18]	An application for full text indexing and searching capabilities
Apache solr[19]	A full-text search, hit highlighting, faceted search, near real-time indexing, dynamic clustering, database integration, rich document (e.g., Word, PDF) handling, and geospatial search
Elasticsearch[20]	A distributed, multitenant-capable full-text search engine with a RESTful web interface and schema-free JSON documents
MarkLogic[21]	A NOSQL and XML database

(continued)

Table 2 (continued)

Big data tools	Description[1]
mongoDB[22]	A cross-platform document-oriented database system, JSON-like documents with dynamic schemas
Cassandra[23]	A scalable multi-master database with no single point of failure
HBase[24]	A scalable, distributed database that supports structured data storage for large tables
InfiniteGraph[25]	A distributed graph database

[1] The description retrieved from each tools official website and Wikipedia at http://wikipedia.org

[2] http://ambari.apache.org/

[3] http://avro.apache.org/

[4] http://incubator.apache.org/chukwa/

[5] http://hive.apache.org/

[6] http://pig.apache.org/

[7] http://spark.incubator.apache.org/

[8] http://zookeeper.apache.org/

[9] http://www.actian.com/about-us/#overview

[10] http://hpccsystems.com/

[11] http://orange.biolab.si/

[12] http://mahout.apache.org/

[13] http://keel.es/

[14] http://www.talend.com/

[15] http://www.jedox.com/en/

[16] http://www.pentaho.com/

[17] http://rasdaman.eecs.jacobs-university.de/

[18] http://lucene.apache.org/

[19] http://lucene.apache.org/solr/

[20] http://www.elasticsearch.org/

[21] http://developer.marklogic.com/

[22] http://www.mongodb.org/

[23] http://cassandra.apache.org/

[24] http://hbase.apache.org/

[25] http://www.objectivity.com/

because consumers could pay on pay-per-use basis for cloud storage or cloud computing services on demand.

The ***hybrid model*** composes private and public clouds. This model could connect a private cloud to public cloud through network connection, such as the Internet.

This model has several advantages, which are listed below:

- *Collaboration between cloud computing systems* Often collaboration between two clouds led to emergence of hybrid cloud model. An organization could keep their own cloud security and maintenance, and simultaneously have collaboration with other clouds. This collaboration could be permanent or temporary.
- *Scalability* This model also is useful for extending the scalability of a private cloud computing system, because in case of limited resources at a peak time, a cluster of new resources could be added temporary from another cloud.

5 Cloud Computing Issues

The cloud computing technology is the best option for dealing with big data. However, cloud computing is still nascent state and we still needed to address some major issues. In this section, we review the major cloud computing issues which are shown in Fig. 2 and are based on an IDC Survey in 2009 [49].

When big data costs customers, and a system disaster could cause organizational destruction in the digital age, migration applications and databases from traditional model are difficult to cloud, because:

- migration to the cloud computing system is difficult; Migration requires to redevelop applications, data and sometimes requires to use efficient programming models to save resources as well as resource costs;
- returning data to the IT department is difficult;
- connection is via an unsecured network, such as the Internet;
- cloud vendor administrator users could have an access to users data;
- data warehouse location is transparent to consumers;
- We do not have a cloud computing standard and standard cloud architecture. It causes some big issues, such as different architectures, difficulty with migration data and application to another cloud vendors;
- We do not have any customization in cloud computing systems;
- We do not have a strong Service Layer Agreement (SLA) for customer satisfaction.

Cloud customers need to have a contract with one or more cloud vendor(s)—often one cloud vendor—and they should use the provided operating systems, middlewares, APIs and/or interfaces. Data and application are dependent on the platforms or are provided by cloud vendor infrastructure. This dependency in

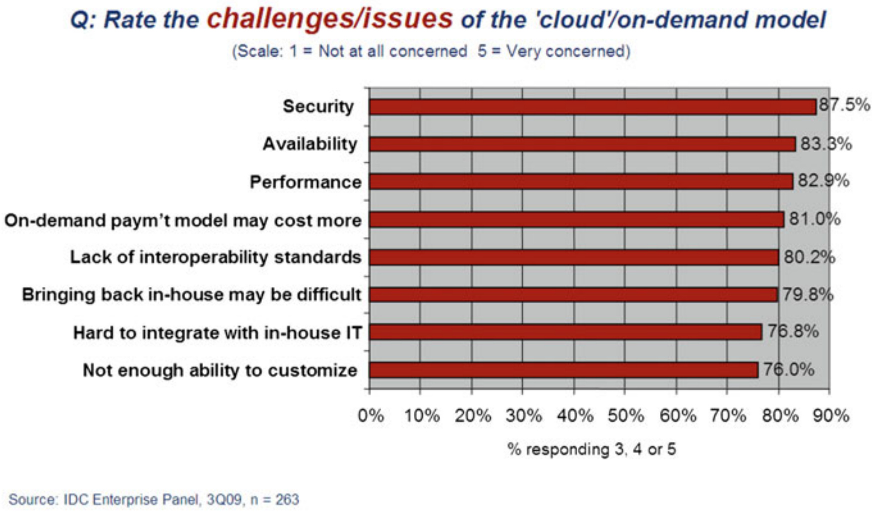


Fig. 2 Major cloud computing concerns [49]

cloud services has several issues. For example, in Fig. 2, "*Security*" is the major concern in cloud computing systems. Cloud features, such as a shared resource pool and multi-user/tenancy cause security issue because the resourced pool are shared through users and we could expose users' data and users' privacy to others.

Unsecured connection to the vendor, network access security, Internet access security and cloud vendors' user security emerged as other major security concerns based on accessibility to the cloud via the Internet.

"*Bringing back in-house may be difficult*" with 79.8 % issue rate and "*Hard to integrate with in-house IT*" with 76.8 % issue rate indicates customers are afraid of data and software application migration to the cloud computing systems, because the migration is difficult to integrate with IT departments and it is difficult to return data back to the IT department; "*Lack of interoperability standards*" with 80.2 % is another cloud issue. This issue shows that cloud computing requires higher interoperability with other cloud computing systems; also as indicated in this report, "*Not enough ability to customize*" with a 76.0 % issue rates show, the cloud computing system requires dynamic architecture and customization.

Some studies, such as [50] show existing cloud computing systems (Amazon EC2 in this case) could not be responsible with a cost-effective performance for HPC applications over using tightly-couple hardware such as Grid Computing or Parallel Computing systems.

To overcome these issues, some study such as [21, 51] are proposed which introduce "Cloud Template architecture". Especially when we employ the cloud computing system for dealing with big data, this architecture is useful. In this study, we show each template could be organized for each purpose and a template could support several service layers simultaneously.

6 Chapter Summary

In this chapter, we discussed a definition of big data, the importance of big data, and major big data challenges and issues. We understand that, if we analyze big data with business intelligence tools, we may provide a catalyst to change an organization to a smart organization. We discussed the importance of cloud computing technology as a solution to handle big data for both computing and storage. We reviewed the capabilities of cloud computing systems that are important for big data, such as resource scalability, resource shrink-ability, resource pool sharing, on-demanded servicing, elastic servicing, and collaboration with other cloud computing systems. We explained cloud architecture service layers and role of each service layer to handle big data. We discussed how business intelligence could change big data to smaller valuable data by using cloud computing services and tools. Finally, we discussed major cloud computing system issues that need to be addressed for cloud computing to become a viable solution for handling big data.

References

1. Matheson, D., Matheson, J.E.: The Smart Organization: Creating Value Through Strategic R&D. Harvard Business Press, Boston (1998)
2. Manyika, J., et al.: Big data: the next frontier for innovation, competition, and productivity (2011)
3. Buscema, M., et al.: Auto-contractive maps: an artificial adaptive system for data mining. An application to Alzheimer disease. Curr. Alzheimer Res. **5**(5), 481–498 (2008)
4. Howe, D., et al. Big data: the future of biocuration. Nature **455**(7209) 47–50 (2008)
5. Hanna, M.: Data mining in the e-learning domain. Campus-Wide Inf. Syst. **21**(1), 29–34 (2004)
6. Wilson, L.A.: Survey on big data gathers input from materials community. MRS Bull. **38**(09), 751–753 (2013)
7. Tan, W., et al. Social-network-sourced big data analytics. IEEE Internet Comput **17**(5), 62–69 (2013)
8. Huang, J., Wu, K., Leong, L.K., Ma, S., Moh, M.: A tunable workflow scheduling algorithm based on particle swarm optimization for cloud computing. Int. J. Soft Comput. Softw. Eng. [JSCSE] **3**(3), 351–358 (2013)
9. Revisited: the rapid growth in unstructured data. Retrieved on 21 Jan 2014 at http://wikibon.org/blog/unstructured-data
10. Infographic: the potential of big data. Retrieved on 21 Jan 2014 at http://blog.getsatisfaction.com/2011/07/13/big-data/?view=socialstudies
11. Taming big data [A big data infographic]. Retrieved on 21 Jan 2014 at http://wikibon.org/blog/taming-big-data/
12. Schonfeld, E.: Google processing 20,000 Terabytes a day, and growing. Retrieved on 21 Jan 2014 at http://techcrunch.com/2008/01/09/google-processing-20000-terabytes-a-day-and-growing/
13. Data, data everywhere. Retrieved on 21 Jan 2014 at http://www.economist.com/node/15557443
14. The big list of big data infographics. Retrieved on 21 Jan 2014 at http://wikibon.org/blog/big-data-infographics

15. Rigsby, J.: Studies confirm big data as key business priority, growth driver. Retrieved on 21 Jan 2014 at http://siliconangle.com/blog/2012/07/13/studies-confirm-big-data-as-key-business-priority-growth-driver
16. Davenport, T.H., Dyche, J.: Big data in big companies, SAS (2013)
17. Fairhurst, P.: Big data and HR analytics. IES Perspect. HR **2014**, 7 (2014)
18. McAfee, A., Brynjolfsson, E.: Big data: the management revolution. Harvard Business Rev. **90**(10), 60–66 (2012)
19. Jacob, A.: The pathologies of big data. Commun. ACM **52**(8), 36–44 (2009)
20. Gewin, V.: The new networking nexus. Nature **451**(7181), 1024–1025 (2008)
21. Bahrami, M.: Cloud computing software architecture and innovation in the cloud. Int. J. Soft Comput. Softw. Eng. [JSCSE] **3**(3), 23–24 (2013). doi:10.7321/jscse.v3.n3.6
22. Young, M.: Automotive innovation: big data driving the changes. Retrieved 26 Jan 2014 at http://www.thebigdatainsightgroup.com/site/article/automotive-innovation-big-data-driving-changes
23. Kelly, J.: Big data in the aviation industry. Wikibon, 16 Sept 203. Retrieved on 18 Mar 2014 at: http://wikibon.org/wiki/v/Big_Data_in_the_Aviation_Industry
24. Siegel, C.F.: Introducing marketing students to business intelligence using project-based learning on the world wide web. J. Mark. Edu. **22**(2), 90–98 (2000)
25. Berner, E.S.: Clinical Decision Support Systems. Springer Science + Business Media, LLC (2007)
26. Marx, V.: Biology: the big challenges of big data. Nature **498**(7453), 255–260 (2013)
27. Liu, F., et al.: NIST cloud computing reference architecture. NIST special publication 500, 292 (2011)
28. Singhal, M.: A client-centric approach to interoperable clouds. Int. J. Soft Comput. Softw. Eng. [JSCSE] **3**(3), 3–4 (2013)
29. Cartier, C., Paynetitle, T.: Optical carrier levels (OCx). Retrieved 24 Jan 2014 (2001)
30. Rittinghouse, J.W., James F.R.: Cloud computing: implementation, management, and security. CRC Press, Boca Raton (2009)
31. Pedrycz, W.: Granular Computing: Analysis and Design of Intelligent Systems. CRC Press/ Francis Taylor, Boca Raton (2013)
32. Bessis, N., et al.: The big picture, from grids and clouds to crowds: a data collective computational intelligence case proposal for managing disasters. In: International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2010 IEEE, New York (2010)
33. Cudré-Mauroux, P., et al.: A demonstration of SciDB: a science-oriented DBMS. Proc. VLDB Endowment **2**(2), 1534–1537 (2009)
34. Bargiela, A., Witold, P.: Granular Computing: An Introduction. Springer, Berlin (2003)
35. Xu, M., et al.: Cloud computing boosts business intelligence of telecommunication industry. In: Cloud Computing. Springer, Berlin Heidelberg, pp. 224–231 (2009)
36. Zorrilla, M., García-Saiz, D.: A service oriented architecture to provide data mining services for non-expert data miners. Decis. Support Syst. **55**(1), 399–411 (2013)
37. Accorsi, R.: Business process as a service: chances for remote auditing. In: IEEE 35th Annual Computer Software and Applications Conference Workshops (COMPSACW), 2011. IEEE, New York (2011)
38. Hunger, J.: Business Intelligence as a Service. GRIN Verlag (2010)
39. Tsai, W.-T., Li, W., Sarjoughian, H., Shao, Q.: SimSaaS: simulation software-as-a-service. In Proceedings of the 44th Annual Simulation Symposium (ANSS '11). Society for Computer Simulation International, San Diego, CA, USA, pp. 77–86 (2011)
40. Candea, G., Stefan, B., Cristian Z.: Automated software testing as a service. In: Proceedings of the 1st ACM symposium on Cloud computing. ACM (2010)
41. Chen, Y., Du, Z., García-Acosta, M.: Robot as a service in cloud computing. In: Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE), 2010 IEEE, New York (2010)

42. Itani, W., Ayman, K., Ali C.: Privacy as a service: privacy-aware data storage and processing in cloud computing architectures. In: Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2009. DASC'09. IEEE, New York (2009)
43. Foster, I., Tuecke, S.: Describing the elephant: the different faces of IT as service. Queue **3**(6), 26–29 (2005)
44. Stanik, A., Matthias, H., Odej, K.: Hardware as a service (HaaS): the completion of the cloud stack. In: 8th International Conference on Computing Technology and Information Management (ICCM), vol. 2. IEEE, New York (2012)
45. Curino, C., et al.: Relational cloud: a database-as-a-service for the cloud (2011)
46. Truong, H.-L., Schahram, D.: On analyzing and specifying concerns for data as a service. In: Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific. IEEE (2009)
47. Zibin, Z.; Jieming, Z., Lyu, M.R.: Service-generated big data and big data-as-a-service: an overview. In: IEEE International Congress on Big Data (BigData Congress), 2013, p. 403, 410, 27 June 2013–2 July 2013
48. Doelitzscher, F., et al.: Private cloud for collaboration and e-Learning services: from IaaS to SaaS. Computing **91**(1), 23–42 (2011)
49. IDC Enterprise Panel, 3Q09. Retrieved on 13 Oct 2013 at http://blogs.idc.com/ie/?p=730
50. Juve, G., et al.: Scientific workflow applications on Amazon EC2. In: 5th IEEE International Conference on E-Science Workshops, 2009. IEEE, New York (2009)
51. Bahrami, M.: Cloud template, a big data solution. J. Soft Comput. Softw. Eng. **3**(2), 13–17 (2013)

Big Data Storage Techniques for Spatial Databases: Implications of Big Data Architecture on Spatial Query Processing

Roger Frye and Mark McKenney

Abstract Today, a large amount of data is being collected and stored. Data that has grown beyond traditional data management solutions has come to be known as big data. Solutions such as Hadoop have emerged to address the big data problem. However, spatial data presents its own challenges to storage and processing. Researchers have taken various approaches with Hadoop to handle spatial data efficiently. The approaches includes multi-stage map/reduce algorithms, generating on-demand indexes, and maintaining persistent indexes. This paper reviews the various approaches, categorizes the spatial queries reported in the testing, summarizes results, and identifies strengths and weaknesses with each approach.

Keywords Spatial data · Big data · MapReduce · Query processing

1 Introduction

Spatial data systems are ubiquitous, and the use of spatial data and spatial systems grows continuously. The traditional landscape of spatial systems is dominated by geographic information systems (GISs) that utilize spatial databases as a storage and processing engine for user interfaces driven by mapping. These systems are traditionally characterized by analyzing relatively static data that are combined and manipulated to create maps. Newer trends in spatial systems reflect a much more dynamic scenario being driven by the decreasing cost and increasing availability of data collection devices. Smart phones create GPS trajectories, geo-tagged

R. Frye · M. McKenney (✉)
Southern Illinois University, Edwardsville, IL, USA
e-mail: marmcke@siue.ed

R. Frye
e-mail: rfrye@siue.edu

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_14,

photographs, and geo-tagged text (e.g., tweets, text messages, social network posts). Remote-controlled drones equipped with cameras can take large numbers of high-resolution photographs that are geo-tagged with the photographs' boundaries so that point in polygon queries can be computed on the images to retrieve images of interest from a database. Temperature sensors, pollution sensors, light sensors, etc. create streams of rapidly updating data over large geographic areas. At the intersection of this area of new, dynamic, and rapidly growing data and the area of big data lies big spatial databases.

Just as spatial databases presented a challenge to traditional data management techniques, big spatial databases present challenges that require trade-offs when designing big spatial data management techniques. Big data solutions, such as NoSQL databases, and distributed file systems, such as Hadoop, provide the ability to scale in a manageable manner and store large volumes of data that grow with high velocity; however, spatial data analysis requires operations that do not always fit the default storage policies provided in such systems. For example, a nearest neighbor query can be efficiently computed using a spatial index structure such as an R-Tree [16], but is computationally costly on a distributed file system where nearby points may be stored on different nodes. This example suggests that spatial decomposition techniques should be materialized physically over a collection of nodes such that points are stored on a node with other points that are nearby. Conversely, a point in polygon query, in which all polygons containing a query point are returned, will achieve very little parallelism in such a materialized spatial decomposition scheme since all the polygons that are near to each other and the query point will be on a single node. This second example suggests that some other spatial decomposition technique, or none at all, should be applied to nodes in a distributed file system supporting a big spatial database.

Big data has implications for decision support systems, automated analysis and computational intelligence. When applied to spatial and spatiotemporal problems, effective and fast analysis can be used in robotics, vehicle and UAV navigation, and agent comprehension surroundings.

In this paper, we present a survey of current research on big data systems that focus on managing spatial data. Our focus will be on systems around the MapReduce framework; because it is a general platform that can be expanded and because it is garnering interesting in the literature. Several projects have taken advantage of Hadoop's scalability and begun implementing spatial queries in a MapReduce scheme. Other researchers have taken to altering Hadoop's default storage distribution to impose a spatial-based distribution of the data in an effort to improve processing performance of such vast quantities of data. One such alteration has been to utilize Z-ordering [15] of the data to place data considered to be spatially close in relatively close proximity in the data storage nodes. Other researchers have imposed an R-tree decomposition, or variants of R-tree, to force Hadoop's Distributed File System (HDFS) to maintain a non-randomized storage structure.

While these projects have the intent to improve the performance of large amounts of spatial data, each researcher, or research team, presents results for specific queries; for example polygon containment, joins, intersections, skyline, etc. The result is a scattering of queries over differing storage architectures with little overarching theme emerging.

The current state of the research leaves open 2 questions: (1) Do specific spatial queries benefit from data organization methods in a distributed file system framework such as HDFS and (2) Do data organizations exist that provide good performance for queries in general? To answer these questions, we perform an analysis of research literature in the area and:

1. Collect queries discussed and/or evaluated in the literature.
2. Categorize those queries.
3. Consolidate query performance results across papers.
4. Analyze the consolidates results.
5. Summarize the implications of current research.

The structure of the paper is as follows. Section 2 categorizes queries that are discussed in the literature. The architecture of the MapReduce based systems will be presented in Sect. 3. Section 4 presents the results of the different query categories for each of the architectures presented followed by implications of those results in Sect. 5. Work that is related but not the direct focus of this review is presented in Sect. 6. Section 7 concludes the paper.

2 Queries

In this section, we introduce spatial data types and spatial queries evaluated by specific MapReduce systems in the literature [1–5, 12, 13, 17, 18, 26, 27, 29–31]. We assign queries to 5 broad categories: Spatial Selection, Nearest Neighbor Variations, Joins, Spatial Aggregation and Feature Aggregation.

2.1 Spatial Data Types

The standard set of spatial data types consists of the types of *point*, *line*, and *region*. Types are further classified as being either *simple* or *complex*. We assume spatial objects are embedded in euclidean space, a typical assumption for many spatial applications. A simple point is then a single coordinate in euclidean space. A simple line is a continuous line with two end points, essentially a connected one dimensional object embedded in two dimensional space. A simple region is defined [23] by a boundary forming a single closed cycle that separates the interior of the region from the exterior of the region, i.e., a Jordan curve. In this paper, we will use the term *polygon* to refer to simple regions with boundaries defined by straight line segments. Figure 1 depicts examples of the simple types.

Simple spatial types have problems in that they cannot represent all of geographic reality and are not closed under all spatial operations. Complex types

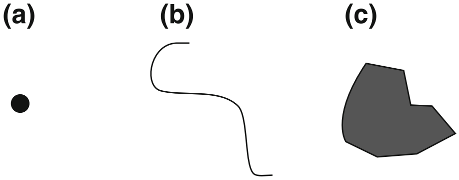


Fig. 1 A simple point (**a**), a simple line (**b**) and a simple region (**c**)

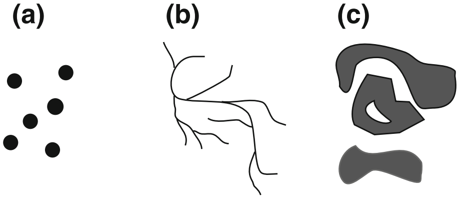


Fig. 2 A complex point (**a**), a complex line (**b**) and a complex point (**c**)

alleviate these problems. A complex point object is a collection of one or more disjoint simple points. A complex line is a collection of possibly disjoint line networks. A complex region can have multiple *faces*, each containing zero or more *holes*. For example, Italy can be represented by a complex region with its mainland and islands forming faces, and hole in the mainland where Vatican City lies. See [23] for formal definitions of spatial types; they are omitted here due to space limitation and are beyond the scope of this paper. Examples of complex types are shown in Fig. 2.

2.2 Spatial Selection

In the reviewed literature that addresses spatial data storage using MapReduce, many authors [1, 4, 13, 17, 26, 29, 31] discuss spatial selection queries. The typical spatial selection query involves finding all spatial objects that are contained within a query region, denoted *range queries*, or that contain a specified query point, denoted *point queries*.

An example range query might include "Find all counties that are contained by a supplied query region". Assume that a relational spatial database has a table X with a column named geom that contains a region. A user provides a query region named "inputRegion". Therefore a query can be formulated as:

```
Select X.geom
From X
Where contains(X.geom, inputRegion)
```

The above query is a spatial selection that returns unmodified objects from the database based upon spatial operations in the "where" clause. If we generalize the query, we identify 2 structures of selections involving frequently used operators.

The first structure uses the operation in the Where condition to restrict the output. This operation can be a topological predicate or spatial set operation in a larger boolean expression, such as contains or intersects.

```
Select [Attribute and spatial column with no operation]
From [tables]
Where operation(geometry, input_geometry)

[additional operators]
```

For the second structure, the operation is used in the Select clause. This structure can optionally include the Where condition.

```
Select operation(X.geom, inputRegion),
[additonal columns and operators]
From X
[Where < expression >}]
```

The key difference between the 2 structures is that the first example will return the original objects within the data set with its specified attributes and spatial geometries. In the second structure, the query will return newly defined regions defined by the operation. These two structures could also be combined.

Many authors utilized region queries for their illustration of selection queries in their evaluations, though some use differing terminology to represent the same type of query. For example, the term “range queries” is used in [13, 17, 27]. Similarly in [29], Zhang states that the more generalized region queries are potentially common queries for large spatial data sets for evaluation purposes, but uses rectangle queries, a type of range queries, like the other researchers. Lastly, in [4], Aji et al. refer to region queries as containment queries. In the Aji research, the query description is left as a more general region query where a polygon is used to define the range rather than a rectangle, as in the other evaluations. As is evident from the list above, range queries are a common query used for evaluation purposes.

In addition to range queries Liao [17] and Zhang [29] use point queries. The point query is to find all regions containing a query point. These queries are also used in the performance analyses.

2.3 Joins

A spatial join performs a spatial operation against two datasets. A Cartesian product is performed between the two datasets where each object in the first set, R, is matched with each object from the second set, S, to see if the pair match some specified condition based upon their geometries. Spatial joins can be data- and compute-intensive operations. As such, spatial joins are examined in much of the reviewed literature to utilize the distributed storage and processing power of MapReduce[1–3, 13, 29–31].

An example query that would require a spatial join would be “Find roads that cross rivers”. To answer this query, a Cartesian product would have to be performed between the set of rivers and the set of roads to see if a pair of objects, represented by complex lines, intersect. Assume we have a relational spatial database with a table Rivers that holds records of rivers and the column geom holds the spatial data representing the river. Similarly, there is a table Roads that contains a record for each road. The query would be written as such:

```
Select Roads.geom, Rivers.geom
From Roads, Rivers
Where intersect(Roads.geom, Rivers.geom);
```

2.4 Nearest Neighbor Variations

The Nearest Neighbor Variations category groups queries that seek to find objects based upon distance. A common nearest neighbor query, referred to as a basic kNN query may ask, “Find the closest point, in a set of points, to a query point”. Variations of the kNN query are discussed in more detail below. Other queries grouped in this category include Closest Pair, which asks the question, “Find the closest pair of points from the set of points” or the inverse, the Farthest Pair, asks to “Find the pair of points from the set of points” that the farthest apart.

The k Nearest Neighbor query (kNN) is a common query type that is frequently discussed in the reviewed literature. There are two sub-types of kNN queries. The basic kNN query specifies a query point and a value for k that will be used to determine the k nearest neighboring objects to the query point. A simplified version with $k = 1$ will find the nearest neighbor. This basic kNN query is used in [5, 13, 17, 31]. The more complex variation of kNN finds the k nearest neighbors for all objects in one dataset R from dataset S. The query will return a list of nearest neighbors found in S for each individual object in R. This query is often referred to as a kNN join, as it requires a spatial join of the datasets R and S. The kNN joins are discussed and often used for evaluation purposes in [1–4, 18, 28]. Both types of kNN queries are mentioned in [29].

There are variations of the kNN query such as Reverse Nearest Neighber (RNN), Maximum Reverse Nearest Neighbor (MaxRNN) and All Nearest Neighbors (ANN). RNN seeks to find all points that have a specific point as its nearest neighbors. It should be noted that while a point p might be the nearest neighbor to a query point q, that does not mean that q is the nearest neighbor to p (See examples below). The MaxRNN query is an extension to RNN that attempts to maximally find a point that has the most points that claim it as its nearest neighbor. The ANN query is actually a kNN join query where $k = 1$; it seeks to find the nearest neighbor for every point within a data set from a second dataset. Figure 3 provides a sample set of points used to further explain the the basic kNN and RNN queries. With the sample data set shown in Fig. 3, if a query were to find the 4 nearest neighbors (kNN with $k = 4$) for query point p, the resulting set would be $\{C, D, G, F\}$. However if the query were to find all the points that have query point p as their nearest neighbor, an RNN query, the resulting set would be $\{D, H\}$. As can be seen, point G would not be in the set because G's nearest neighbor is point F. A similar situation applies to point C with nearest neighbor B. Point H did not qualify as one of the 4 nearest neighbors to point P; However, point H does have point P as its nearest neighbor.

Most of the literature in this survey used kNN and/or kNN join sample queries in their research. However, Zhang [29] and Wang [26] discuss the ANN query, while only Akdogan [5] defines the RNN and MaxRNN variations.

While Eldawy did use the kNN query to demonstrate SpatialHadoop [13], two alternate variations of the nearest neighbor query are used for evaluation in [12]. The Closet Pair query, seeks to find the pair of points with the shortest distance between the two points. In Fig. 3, the Closest Pair query would result in $\{G, F\}$. The Farthest Pair finds the pair of points with the greatest distance between the two points. Points $\{A, I\}$ from Fig. 3 match this query. The kNN join, Closest Pair and Farthest Pair queries can be solved using a Cartesian product for each point with every other point but they have not been classified as join queries as they specifically use a distance metric.

2.5 Spatial Aggregation

Spatial aggregation queries determine the solution to a query that requires combining geometric characteristics of objects. For example, a spatial aggregation query might ask to "Find the average area of all counties", where county boundaries are represented by polygons. The query must take the average of all polygons that represent counties:

```
Select avg(area(Counties.geom))
From Counties;
```

In [12], Eldawy utilizes a collection of spatial queries that we categorize as spatial aggregation queries. The queries, Union, Skyline and Convex Hull produce a single object as the result.

An example Union query might ask "Find a single region encompassing the area of all counties with a population of 100,000". For this Union query, the result will be a newly created complex region object whose boundary contains the union of the interiors of all polygons that meet the query criteria.

The Skyline query is formally defined as a spatial operator in "The Skyline Operator" [9]. Eldawy defines the Skyline queries for his use as "The skyline of P consists of those points of P that are not dominated by any other point of P" A point dominates another point by being *greater* in one dimension of its coordinates. Using the dataset in Fig. 3, a Sklyine query would result in points $\{A, B, E\}$. By examining Fig. 3, it is apparent that point A is not dominated by any other point

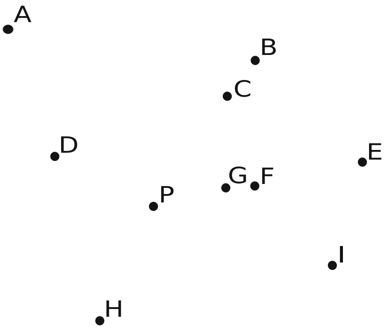


Fig. 3 A point set for kNN and RNN examples

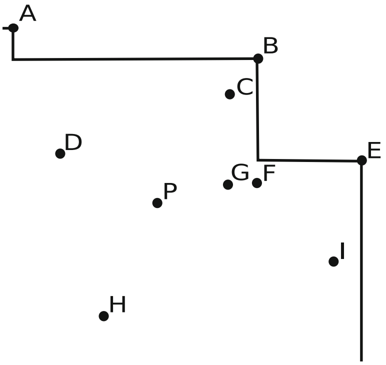


Fig. 4 The results of a Skyline query on the points depicted in Fig. 3

because its *y* coordinate is greater than all other points. In much the same manner, point *E* is not dominated by any other point because its *x* coordinate is greater than all other points. Point *B* is not dominated by any other points. For any point left of *B*, *B* dominates those points with regard to x coordinate. For points to the right of *B*, *B* dominates those points in the *y* dimension. Thus, *B* is not dominated by any other point. With {*A*, *B*, *E*} shown connected in Fig. 4, it is evident how the query gets its name.

For a Convex Hull query, the smallest polygon (simple region) containing a set of query points is computed and returned [12]. Using the dataset from Fig. 3, the resulting set of points from a Convex Hull query would be {*A*, *B*, *E*, *I*, *H*} as shown in Fig. 5.

2.6 Feature Aggregation

As with any data storage, a user may want to query against any characteristic or feature preserved in that data. Even though a data store is built and based around spatial data, those spatial objects my have associated non-spatial features that users

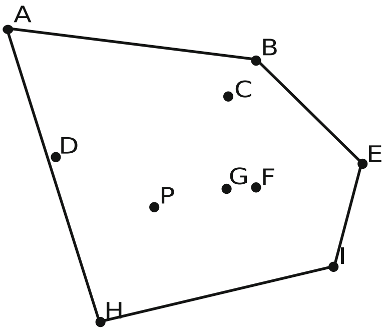


Fig. 5 The result of a Convex Hull query on a set of query points

may want to collect and present. Feature aggregation for non-spatial features is still a potential type of query to be performed on what is fundamentally spatial data. Additionally, feature aggregation queries can be integrated with many of the spatial queries. For example, you may want all objects with a specific feature but only within a selected spatial region. These types of queries are not our focus, but included for completeness. Aji et al. [3] and Wang and Wang [27] mention non-spatial feature aggregation queries in their research.

3 Architecture

In this section, we discuss the fundamentals of Hadoop, followed by a discussion of the architectural approaches using Hadoop to store and process spatial data. There have been 3 approaches to adopting MapReduce for spatial data. Some projects such as [5, 18, 27, 29, 30] focus on developing algorithms to utilize the distributed power of native Hadoop without modification, implying that no indexes are imposed on the data store other than what Hadoop may use. Others [1–4, 27] have taken an approach to generate temporary indexes on-demand to aid in the query performance. Others still, [12, 13, 17, 31] have added a software layer atop HDFS that organizes physical nodes into an indexing scheme to improve the efficiency at locating spatial data.

3.1 Hadoop Framework

In 2004, Google engineers published a paper on Google Filesystem (GFS) [14], followed in 2004 with a paper on MapReduce [11]. These papers describe a distributed computing framework for storing and analyzing massive amounts of data. The framework described in these papers prompted the creation of an open source implementation of MapReduce and GFS, called Hadoop. The primary purpose behind Hadoop is storing what has become known as Big Data and the ability to run analysis on that data.

Algorithm 1: A sample nearest neighbor query using the map/reduce structure.

```
Input: A set of points $P$ assumed to be on a distributed file system. A query point $p$.
Output: The point $p \in P$ nearest to $q$ based on euclidean distance.
map(Key key, Value q):
    for each value:
        EmitIntermediate(0, q)

Reduce(Key key, Value qArray[]):
    d = distance(p, qArray[0]);
    closestPoint = qArray[0]
    for each q in qArray:
        if (distance(p, q) ¡ d):
            d = distance(p, q)
            closestPoint = q
    emit(0, closestPoint)
```

Hadoop has two key components, Hadoop Distributed Filesystem (HDFS), an open-source implementation of GFS, and an open source implementation of MapReduce. Hadoop stores large data sets using HDFS in a master-worker configuration with one (or more) machines acting as a master node and several other nodes as worker nodes. This data is then analyzed with tasks called MapReduce jobs. MapReduce jobs are divided into two stages. During the Map phase, the input data is processed by mapping a key and value pair to a resulting key and value pair, (k1, v1) → (k2, v2). The output from the Map phase becomes the input to the Reduce phase. The Reduce phase also results in a key-value pair such that (k2, v2) → (k2, v3).

To illustrate how MapReduce could be used for a spatial query, we will use an example query. Suppose we want to perform a basic nearest neighbor query to locate the nearest object to a query point. Let p be that query point. Assume that the Mapper will execute over all points in the data store. A pseudo-code example for a Map/Reduce job would appear as in Algorithm 1.

3.2 Non-indexed Algorithms

As previously mentioned some approaches attempt to devise efficient algorithms in which to perform spatial queries in a Hadoop installation without the use of indexes. In brief, the non-indexed approaches described in the reviewed literature [12, 18, 26, 29, 30] utilize two MapReduce jobs in sequence to perform a single query. Generally, the first MapReduce job is used to perform spatial filtering to reduce the number of objects to be processed. The second MapReduce job will then process the remaining objects to check for matching criteria. Unlike the other two categories, no indexes are imposed on the data, other than any indexing the HDFS system might be using by default.

3.3 Temporary Indexes

Recognizing the benefits of having indexes for spatial data, two projects create indexes on-demand. The first project has evolved in several publications [1–4]. Aji first described a spatially aware implementation using Hadoop in [2], referring to it as Medical Imaging Geographic Information System (MIGIS). The system is described as using only temporary indexes. Although the system was original aimed at vector objects converted from medical images, the system evolved as described in [3] to include more general purpose GIS uses. The latest revision, referred to as Hadoop-GIS, is described in [1, 4]. The system has evolved into a full spatially aware storage and query system with Hadoop at its core. It still uses temporary indexes but in conjunction with global indexes that are stored in HDFS itself. For the purposes of this review, we are categorizing it as a temporary index solution as it still generates indexes on-demand and early evaluations are reported using the temporary-only indexes.

Rather than developing individual spatial query algorithms, the intent for Hadoop-GIS is to be a fully developed spatial query system. As such, it takes a multi-layered approach that provides an interface with its own query language, a translator to convert the query into a MapReduce job or jobs, a query engine that processes each query according to its needs and the storage layer which handles data partitioning and staging across HDFS.

As discussed in Zhang et al. [29, 30] and Wang and Wang [26], a non-uniform distribution of data within the data space, where some areas have a high number of objects while other areas have very few, can cause an unbalanced workload in a distributed Hadoop environment. This problem is referred to as data skew. As such, in Hadoop-GIS [4], partitioning and storage of that data across HDFS is performed before queries are processed. The approach breaks the data space into tiles where each tile holds a maximum number of objects, recursively breaking high density tiles into smaller tiles. Once the data has been partitioned and stored in HDFS, a user can issue queries to the system.

To implement the query interface in Hadoop-GIS, the authors have extended HiveQL to include spatial operators, functions and data types. Hive supports query translation to MapReduce jobs through an SQL-like query language called HiveQL or simply QL [7]. For Hadoop-GIS, the authors extend HiveQL with spatial query operators such as ST_INTERSECTS, ST_CONTAINS, and ST_KNN, along with spatial data types such Point, Polygon, Box, and LineString [1].

Once the query has been translated to MapReduce jobs, a spatial query engine, developed as part of Hadoop-GIS, called Real-time Spatial Query Engine (RESQUE), will take advantage of global tile indexes from preprocessing in the data partition phase and will generate local indexes on demand. It is stated that the overhead of generating the temporary indexes is a "very small fraction of compute- and data-intensive spatial [query]" [4] processing time. These indexes are used to optimize the MapReduce job before submitting it to the underlying Hadoop framework.

3.4 Persistent Index

The final categorized approach utilizes persistent indexes. We define a persistent index to be an index that is not constructed on the fly, but is maintained across multiple queries. For example, each node in a cluster could contain a spatial index for the spatial objects on that node to accelerates queries on that node; we denote such an index a *node level index*. Another example is that data could be physically organized across nodes such that spatial objects, points for example, that are near to each other in their embedding space are stored on the same node; we denote such indexes *physical indexes*. In the reviewed literature [5, 12, 17, 31], all systems that used node level indexes also organized data according to a physical index, therefore we use the term persistent index to refer to systems that use both.

Research projects that have added persistent indexes to Hadoop include Akdogan et al. [5], Eldawy et al. [12, Eldawy and Mokbel 13], Liao et al. [17] and Zhong et al. [31]. Of these projects, only Akdogan uses a flat index. This index is based upon a Voronoi Diagram. All others use hierarchical indexes such as R-tree or Quad-tree indexing. A prime example of a project that integrates a hierarchical index in the MapReduce framework is the open source project SpatialHadoop, which can be found at the SpatialHadoop website [24]. We will describe the architecture of SpatialHadoop as an example or a persistent index system.

In [13], Eldawy describes SpatialHadoop and how it uses persistent indexes to extend Hadoop for use with spatial data. SpatialHadoop allows normal users to perform spatial queries. To these users, the indexes are hidden just as with a traditional DBMS. SpatialHadoop is also extensible which allows developers to add spatial query capabilities.

The basic architecture of SpatialHadoop is comprised of four layers: language, storage, MapReduce, and operations. The language layer is aimed at the end-user for submitting queries. This layer has been implemented as an extension of Pig Latin, an SQL like language [20]. The extension includes support for spatial data types and operations. This SQL-like language is the user interface to issue spatial queries.

The storage layer extends HDFS with the addition of a two-level index structure comprised of global and local indexes. The global index is a physical index (as defined above) and is stored in main memory on the master node of the Hadoop system; it is an index across all nodes in the cluster. The global index is used in the preparation of MapReduce jobs involving spatial queries. Local indexes organize data on each node and are used to process Map tasks. These indexes are stored as one block file per node, which is 64 MB as this is the default block size used by HDFS. Local indexes can be used to support a variety of index types, namely a grid file, R-tree or R+-tree. These indexes themselves are created through a MapReduce job.

The MapReduce layer adds two new components to the basic Hadoop installation, SpatialFileSplitter and SpatialRecordReader. These two components are aimed at helping developers integrate spatial operations into the operations layer (described below) as MapReduce programs. In general, the SpatialFileSplitter utilizes the global index to prune data that is not included in a query solution, while the SpatialRecordReader uses the local indexes to find matching records for the query.

The fourth layer, operations, is where developers implement spatial operations. The initial release of SpatialHadoop has three spatial operations implemented to illustrate how SpatialHadoop supports spatial data. The operations are range query, basic k nearest neighbor, and spatial join. The system is intended to be extended with additional spatial operations.

In [12], Eldawy developed a collection of computational geometry operations collectively referred to as CG_Hadoop. With CG_Hadoop, two approaches were taken, creating algorithms for a native Hadoop installation with no indexes and a version of each operation for SpatialHadoop using its indexes. Thus results from Eldawy's evaluations of CG_Hadoop are presented for non-indexed and persistent indexed architectures.

4 Reported Results

In this section, we summarize the query evaluation results reported in the reviewed literature. The section will follow the query categories, Spatial Selection, Joins, Nearest Neighbor Variations, Spatial Aggregation, and Feature Aggregation, as presented in Sect. 2 while grouping results based upon the architecture: Non-indexed Algorithms, Temporary Indexes, and Persistent Indexes.

4.1 Spatial Selection

These results are compiled from the reviewed literature that discusses and evaluates spatial selection queries in a Hadoop-based environment.

4.1.1 Non-indexed Algorithms

As previously indicated, Zhang et al. [29] developed spatial selection query solutions without indexes in the MapReduce framework. To evaluate the performance, queries were performed using two data sets, both from TIGER [25] census data for 2007. The first set is the collection of roads in California, which contains over 2 million objects for a total size of 529 MB. The second data set is the hydrography features in the state of California. This set contains 373,950 objects for a total of 135 MB of storage. Each query was evaluated using from 1 to 6 Hadoop nodes. For comparison, using the same data set, the query was performed on an installation of Oracle Spatial.

Zhang utilized both range queries and point queries to evaluate the non-indexed MapReduce solutions. For the point query, Zhang states that the MapReduce solution took several seconds to complete while the Oracle Spatial query finished in under a second. This indicates the point query solution does not perform well without indexes.

For the range queries, two window queries were used to test the impact of the size of the query window. One window query used a small window that covered only slightly more than 1 % of the data space. The other window covered more than one-third the size of the data space. The small window query exhibited similar results to the point query. While the execution time did improve for the small window query as more cluster nodes were used, the 6-node Hadoop installation was outperformed by Oracle Spatial. The Hadoop implementation was able to show an improvement when executing the large window query. With the large query windows, the benefit of the indexes on the Oracle database was lost with such a large area involved in the query.

From all of the examples, point query, small query window and large query window, spatial selection queries favor the use of indexes in the Oracle database unless the query covers a large enough area of the data set that the indexes no longer provide significant benefit. In both, the point query and small window query, the use of indexes in the Oracle Spatial environment provides an order of magnitude greater performance over the Hadoop based solution. It is only when the query window covers a relatively large portion of the data set, that the non-indexed MapReduce query outperforms the indexed platform with regards to spatial selection queries.

4.1.2 Temporary Indexes

There are two temporary index solutions that report results for spatial selection queries. The first, Wang and Wang [27] implementation generates indexes on-demand for performing spatial selection, or window, queries in a Hadoop environment. Oracle Spatial is used for evaluating the on-demand index solution on the MapReduce framework. The Hadoop environment is a 10-node cluster storing two data sets. The first dataset contains over 2 million building polygon objects. The second set contains nearly 5,500 roads stored as polylines.

The size of the window in relation to the data space was not provided but was given as 1 km × 1 km. The same window was used to query against each dataset separately. While the Hadoop platform did slightly outperform Oracle Spatial, both platforms performed similarly on the spatial selections.

Additional results are reported for Hadoop-GIS, an on-demand indexing architecture described in [1, 4]. The evaluations of this platform are performed on an 8-node Hadoop cluster. In [1] Hadoop-GIS is compared to an unmodified Hadoop installation. In [4], Hadoop-GIS is compared against an unnamed commercial DMBS running on two nodes, simply referred to as DBMS-X. Two datasets are used in the evaluations. The first dataset contains approximately 400 million spatial objects having 74 attributes. The objects were derived from medical images and require 70 GB of storage. The second set contains several million objects extracted from OpenStreetMap [21], requiring 300 GB of storage.

In [1], four different window sizes are used in the window queries. The size of the windows in relationship to the data space it not given, merely described as S, M, L, XL. The results of the temporary indexed vs the non-index Hadoop solution shows that indexes significantly help for smaller window sizes. As the window size increases, the benefit of the indexes is reduced.

Only one window size is used in [4] to compare an 8-node Hadoop cluster against a 2-node DBMS-X cluster. The window size is not specified. Rather than vary the window size, the evaluations compare partitioning sizes of the dataset. The results show that with proper data partitioning, the DBMS-X can slightly outperform the Hadoop installation. Examining these results in conjunction with those presented in [1], the on-demand indexing benefit spatial selection queries but not enough to outperform a well configured traditional DBMS.

4.1.3 Persistent Indexes

From the reviewed literature, three articles discuss spatial selection queries using persistent indexes integrated into a Hadoop framework. Two of the articles, [13, 17] do not report performance evaluations that are pertinent to this review. In [13], Eldawy is merely establishing a demonstration of the SpatialHadoop environment and has implemented a window query as part of the demonstration. No performance evaluation is reported. In [17], Liao does perform evaluations using spatial selections queries but there is no comparison to a traditional DMBS or another Hadoop environment. The focus of the evaluations is for index configuration options to determine improvements in index construction in the Hadoop environment.

Zhong et al. [31] reports performance evaluation results for a persistent indexed Hadoop installation that is referred to as VegaGiStore. The tests are performed on a 17-node cluster. VegaGiStore is compared against two traditional DMBS. Similar to others, Zhong compares the proposed system to the commercial Oracle Spatial. The third system used for comparison is the open-source PostGIS [22], an extension to PostgreSQL. Both Oracle Spatial and PostGIS are configured to run on the same 17 computing nodes as the Hadoop-based system.

Three spatial queries are performed with a data collection that contains over 314 millions points, over 81 million line objects and over 16 million polygon objects. The data is only described as real spatial map data. Each query is tested with differing number of nodes, from 1 to 17. The first query finds all the points within a specified query window. The second query seeks all lines that are contained or intersect with the specified window. The third query finds polygons that are contained or overlap with the window. In all three cases, the Hadoop installation performs slightly slower on one node. However, the VegaGiStore implementation scales more effectively than the other two implementations. VegaGiStore improves the execution time by an order of magnitude when the queries are run on 17 nodes and shows significantly lower response times than either of the two DBMS systems. The other systems do not scale as well; showing Hadoop with persistent indexes can be used to improve spatial selection queries.

4.1.4 Summary

Of the three architectures, only the persistent indexed implementation shows consistent benefits with regards to spatial selection queries. The reported results from the non-indexed solution showed that the Hadoop solution did not perform as well as a traditional DMBS with the single exception of window queries with a large window size. In these queries, the large coverage area overcame the benefit of the indexes in the traditional DBMS. As the non-indexed solution did not perform well in a majority of the evaluations, point queries and windows queries with small windows sizes, and it is unclear at what point the size of the window query begins to negate the benefit of indexing, the non-index solutions are not a good fit for spatial selections. Similarly, the results for the temporary indexes indicate that this architecture is not well matched for spatial selection query as a well tuned Oracle installation running on fewer nodes was able to match the Hadoop solution. The results presented for VegaGiStore, a persistent index solution was shown to scale better than two traditional DBMS's Oracle Spatial and PostGIS. Thus, only the persistent index solutions are a good solution for the case of spatial selection queries for the data presented in the literature.

4.2 Spatial Joins

This section reports the results from evaluations of spatial joins that were reported in the reviewed literature.

4.2.1 Non-indexed Algorithms

The focus of [30] is developing a non-indexed MapReduce algorithm for spatial joins. As such, the name of the project is Spatial Join with MapReduce, abbreviated SJMR. In [30], the evaluations are shown to illustrate how algorithm alterations impact performance. As such there are no results to compare the effectiveness of the solutions to other architectures. The work was followed up by Zhang in [29].

As mentioned above, this article compares the performance against Oracle Spatial using road and hydrography data. As mentioned above, the evaluations are performed using non-indexed algorithms on an unmodified Hadoop installation using 1–6 nodes. As reported, only in the single-node scenario does Oracle Spatial out perform the SJMR solution with a small margin. With the addition of a second node, SJMR executes the query in just over half the time of the Oracle Spatial implementation. As nodes are added, the performance continues to increase for SJMR showing significant benefit over a traditional DBMS, as well as scalability.

4.2.2 Temporary Indexes

As reported above, Wang and Wang [27] used temporary indexes in a 10-node Hadoop cluster which was evaluated against Oracle Spatial. The reported results include a single spatial join query that finds intersecting roads from the approximate 5,500 road dataset. The reported times show that, even with the overhead of generating indexes on-demand, the Hadoop solution performs the query in less than half the time of the Oracle Spatial platform.

Spatial Joins are discussed throughout the works written by Aji [1–4]. For our purposes, results are reported in [4] for a system that evolved into an on-demand index solution called Hadoop-GIS. The evaluations are performed on an 8-node cluster with 192 cores. Hadoop-GIS is compared to a commerical parallel Spatial DBMS that is simply referred to as DBMS-X. Here it is shown that both Hadoop-GIS and DBMS-X scale well. In all cases shown, Hadoop-GIS outperforms DBMS-X. Aji indicates that one reason the Hadoop-GIS system works well is that the on-demand indexing algorithm is better at handling data skew.

4.2.3 Persistent Indexes

Of the reviewed literature, three articles discuss the implementation of spatial joins using persistent indexes integrated into a Hadoop framework. In [13], Eldawy does show that SpatialHadoop has incorporated the ability to perform spatial join queries but does not report any evaluation results. As previously stated, Liao [17] only uses the evaluation queries to improve the index configuration.

As with spatial selection queries, Zhong [31] performs evaluations on a 17-node cluster comparing VegaGiStore, a Hadoop-based framework that uses persistent indexes, with Oracle Spatial and PostGIS. The spatial join query used seeks to find lines that intersect within two datasets with over 24 million lines each. This query was executed using 1–17 nodes. Just as with the spatial selection results, VegaGiStore is slightly slower than the other two systems when the query is run on one node. Yet again, VegaGiStore scales much more rapidly, by more than an order of magnitude, when using the 17-node deployment. The other implementations do no scale as well. As with the spatial selection, a persistent index in a Hadoop framework is a good choice for spatial joins.

4.2.4 Summary

Based upon the reported results, Hadoop is an effective platform for performing spatial joins. All three architectures, Non-indexed Algorithms, Temporary Indexes and Persistent Indexes, were reported to outperform traditional DMBS's.

4.3 Nearest Neighbor Variations

In this section, results for queries categorized as a variation of the nearest neighbor queries are summarized. As stated earlier, there are two different, fundamental kNN queries. The basic kNN query seeks the k nearest neighbors to a specified query point. The more complex kNN query is a kNN join and seeks to find the nearest neighbors for each object in R from data set S. Other queries that are grouped here and discussed in the reviewed literature include, ANN, RNN, MaxRNN, Closest Pair and Farthest Pair.

4.3.1 Non-indexed Algorithms

There are three solutions presented that uses nearest neighbor queries to show performance results for non-indexed algorithms. There is insufficient data reported for CG_Hadoop. In [12] a collection of spatial queries that are executed on two architectures, an unmodified, hence non-indexed, Hadoop installation and an indexed implementation called SpatialHadoop that is described in the Architecture section. Of this collection of queries, Closest Pair and Farthest Pair have been categorized as nearest neighbor variations. However, there are insufficient results reported for either of these queries executed on the non-indexed Hadoop installation. There are results discussed below for Closest Pair and Farthest Pair executed on SpatialHadoop, an indexed Hadoop implementation.

In [26], Wang uses the ANN query to compare a non-indexed MapReduce algorithm to the performance of the same query on a single node Oracle Spatial machine. The MapReduce algorithm is run on 2, 4, 6 and 8 node configurations.

For the experiment, two types of data, real and synthetic, are used for evaluation purposes. The collection of real data is extracted from TIGER data files. Two sets of TIGER data are used, one contains over 2 million line objects representing roads in California and the other approximately 373,000 lines representing rivers in California. With this data collection, a query to find all the roads nearest to each river as well as the inverse, a query to find all rivers nearest to a road are issued. The synthetic data collection is comprised of three random sets of points with 4, 1 million and 100,000 points. Similar ANN queries are issued for the random point datasets.

The single node Oracle Spatial performs as well and in most cases better at executing the ANN queries then when they are executed on a 2-node Hadoop cluster. It is only with 4 nodes and beyond that the Hadoop algorithms perform better than the traditional DBMS. It should be noted, that all comparisons are done with a single-node Oracle implementation.

Results from a third solution are presented by Lu in [18]. For the evaluations, a 72-node Hadoop installation is used. This configuration is compared against an indexed Hadoop solution that uses R-trees. The R-tree installation used here for comparison is discussed in [28] (which is mentioned in the Related Work section below). For the experiments, real forest data taken from [8] to create 2 datasets. The real forest data includes 580,000 objects with 54 attributes. Additionally, the dataset was expanded by creating synthetic data based on the real data to increase the number of objects from 5 to 25 times. The third data set is a collection of 10 million points taken from OpenStreetMap [21].

The focus for the solutions presented by Lu are for kNN join queries. For evaluations kNN join queries were issued for various values of *k*, using various dataset sizes, and a varying number of computing nodes. The results from all performed tests indicate that the non-index solution has slightly shorter run times than the indexed solutions. Uses differing values {10, 20, 30, 40 and 50}, for *k*, the tests show that the non-index solution executed the queries slightly faster for both the artificially expanded forest data with 5.8 million objects and the 10 million real data points from OpenStreetMap for all values of *k*. Tests were performed to evaluate the scalability of the solution using varying sizes of the artificial forest data ranging from the 5 to 25x the real size in increments of 5. As the size of the data is increased the running times increased. The running time for the indexed solution increased in a linear fashion while the non-indexed solution did not increase as much as the data size increase showing better performance as the data size increased. Additionally, tests were run to measure the effect of the number of nodes ranging from 9 to 36 nodes. Again, the non-indexed solution perform slightly better than the indexed solutions.

Based upon the comparison to Oracle Spatial by Wang and the comparison to an indexed Hadoop solution by Lu, there is evidence that non-indexed solutions can be developed to efficiently support nearest neighbor type queries.

4.3.2 Temporary Indexes

None of the reviewed literature that implemented temporary indexes provided comparative evaluation results. While Aji does discuss kNN join queries in [2, 3], the experimental results are used to verify that the solutions do scale as more nodes are added. Therefore, no conclusions can be drawn about the performance of temporary indexes in Hadoop against other solutions.

4.3.3 Persistent Indexes

There are four implementations of persistent indexes that discuss nearest neighbor queries.

In [12], Eldawy demonstrates the effectiveness of SpatialHadoop with regards to the Closest Pair and Farthest Pair queries. The queries were issued on a 25-node SpatialHadoop cluster and compared to a single machine running the same algorithm on a non-Hadoop system. For the experiment, generated synthetic data was used to create a large dataset of points. The results of the Farthest Pair query show that the single machine significantly outperforms SpatialHadoop when used on 4 and 8 GB of generated data. The results are flipped for the Closest Pair query showing SpatialHadoop outperforming the single machine. The most significant conclusion shown by the results presented is that SpatialHadoop scales as the dataset scales because the single machine implementation was unable to process datasets larger then 8 GB while the SpatialHadoop cluster was shown to process datasets up to 32 GB in size. Effectively the only conclusion to be reached for SpatialHadoop with regards to nearest neighbor queries is that it scales to handle larger datasets.

Liao et al. [17] used basic kNN queries to evaluate configuration options for best utilization of the indexes. Just as in the spatial selection query evaluations performed by Lu (described above) no conclusions can be made about the performance against other implementations executing nearest neighbor queries.

The basic kNN query to find the nearest objects to a query point is used by Zhong et al. [31] to evaluate VegaGiStore, a persistent index MapReduce system. As with the spatial selection queries, Zhong compares VegaGiStore against Oracle Spatial and PostGIS. The kNN queries are issued twice, once using $k = 1$ and the other $k = 10$. Both kNN queries are executed across multiple nodes, from 1 to 17. Similar results are observed with the basic kNN queries as discussed above with the spatial selection queries. VegaGiStore has longer response times on a single node but scales much more quickly than the other two systems for both values of k. Additionally, the VegaGiStore implementation shows an even greater speed up when $k = 10$, such that the performance nearly matches the performance for $k = 1$. As such, queries were performed on VegaGiStore only to increase k from 10 to 55 in increments of 5. The results shown indicate that the kNN query time does not increase significantly as the value of k is increased. These results indicate that an indexed Hadoop installation such as VegaGiStore is efficient at performing kNN spatial queries.

Of the indexed approaches, Akdogan is unique in that it is the only flat-index, as opposed to a hierarchical index such as R-tree, that is based upon a Voronoi Diagram constructed from the points of a dataset. This Voronoi Diagram-based index is also created through a MapReduce job, as is the case for the R-tree based indexes. Akdogan indicates that the process for creating the index requires more time than constructing an R-tree index. But, the construction process does scale as more nodes are added to the cluster.

To evaluate the query performance, Akdogan performs a series of tests that includes ANN, RNN, MaxRNN and kNN join. Although the setup and comparator for each query varies slightly, each query was performed on 1–6 Hadoop nodes. The datasets for the test contain real data from Navteq and includes a set of 450,000 points representing restaurants and a set of 1,300,000 points representing businesses from the entire United States. The ANN query was performed using both sets of points. In each case, the query was run on 1–6 Hadoop nodes. The results show that the Voronoi based solution scaled better than the R-tree solution and produced 5 times faster response times. Both RNN and MaxRNN were not compared against MapReduce solution but the Voronoi-based solution was shown to have good scalability as nodes were added to the cluster.

Finally, Akdogan uses a kNN join query to compare the Voronoi-based solution with a solution presented by Zhang in [29], which is a non-indexed MapReduce architecture discussed above. For the evaluation, the values of 1, 5, and 10 are used for k. Additionally, multiple kNN queries, 100, 500, and 1,000, are submitted to both architectures. With both sets of test, varying k and increasing number of queries, the Voronoi-based solution dramatically outperforms the non-indexed solution.

4.3.4 Summary

Although, the queries we have categorizes as Nearest Neighbor Variations are the most commonly discussed in the reviewed literature, results from temporary indexed solutions were not presented. Examining results from the non-indexed and indexed architectures shows that both methods are beneficial to the performance of nearest neighbor based queries when used in the MapReduce framework. Nothing can be concluded about the performance of temporary indexed solutions as there is insufficient data reported with these types of architectures.

4.4 Spatial Aggregation

Of all the literature reviewed, only two articles, [4, 12] included any discussion specifically regarding Spatial Aggregation queries.

4.4.1 Non-indexed Algorithms

In [12], Eldawy presents a collection of routines, titled CG_Hadoop. The queries, Union, Skyline and Convex Hull, are each implemented in a non-index Hadoop and indexed Hadoop framework. To preserve uniformity in reporting, only the indexed results are presented here.

The Union query in CG_Hadoop is performed using various sized subsets (0.25, 1, 4, and 10 GB) of real polygon data extracted from OpenStreetMap. The polygons represent lakes and parks. The single machine implementation fails on the 4 and 100 GB datasets. The unmodified Hadoop cluster outperforms the single machine in both cases but also continues to handle the larger datasets.

Evaluation for the Skyline query are performed on multiple datasets including real and synthetic data. The real dataset is a collection of 1.7 billion points representing points of interest from around the world that was extracted from OpenStreetMap. The non-indexed execution performed the Skyline Query 8x faster than the single machine implemention. The query was also performed using data sets of various sizes (4, 8, 16, 32, 64 and 128 GB) of randomly generated points using Uniform, Gaussian, Correlated and Anti-correlated distributions. In all cases, the non-indexed Hadoop installation achieves an order of magnitude improvement over the single machine.

The Convex Hull operation was also performed on the same real dataset as the Skyline query and the synthetic datasets. It also exhibited an 8X speedup over the single node machine in the case of the real dataset and and is described as "much faster than the single machine" [12] for the synthetic data.

Although the evaluation for Convex Hull query is cut short due to memory limitations in the single machine implementation the results do show that the non-index Hadoop solution scales well. Combine those results with the results shown for Skyline and Convex Hull operations, the non-indexed solution does perform well for Spatial Aggregation queries.

4.4.2 Temporary Indexes

As described above, Aji [4] evaluated the performance of Hadoop-GIS against a commercial database referred to as DBMS-X. Unlike the results for Joins and Spatial Seleciton queries, Hadoop peformed poorly against DBMS-X. Solely based upon this one example, there is no evidence to indicate that a temporary indexed MapReduce solution is adequate for Spatial Aggregation queries.

4.4.3 Persistent Indexes

As mentioned earlier in this section, Eldawy [12] implemented the collection of spatial operations called CG_Hadoop in non-indexed and indexed variations. The non-indexed results for the Union, Skyline and Convex Hull operations are reported above. Each performed well in the unmodified Hadoop cluster when compared to a single machine implementation. The indexed, SpatialHadoop implementation showed even greater speedup for each query with both real and synthetic data. Some speed-ups were very dramatic. In the case of Skyline and Convex Hull operations performed on the real data, the speed-up was 80x over the single machine. Recall that the non-indexed solution showed an 8x speed-up for these two operations. When executed on the synthetic data, Skyline achieved two orders of magnitude performance improvement on SpatialHadoop and Convex Hull exhibited a 260x speedup for the largest dataset. As the non-indexed solutions were shown to be well suited for Spatial Aggregation operations, the persistent indexed solutions achieved far greater benefits.

4.4.4 Summary

For spatial aggregation queries, the collected results rely upon only two evaluations for all three architectures. The implementation of the CG_Hadoop collection of operations on both non-indexed Hadoop and SpatialHadoop is the sole source of results regarding spatial aggregation. Based upon the results presented, both non-indexed and indexed solutions of SpatialHadoop process spatial aggregation well. It is clearly shown that SpatialHadoop, the indexed solution, perform significantly better than the unmodified Hadoop platform. The only source reporting results for a temporary indexed solution is with Hadoop-GIS. The reported results show poor performance when compared to a traditional DMBS. Based upon the limited evaluations, both non-indexed and indexed solutions perform well on spatial aggregation queries while the on-demand index solution does not perform well.

4.5 Feature Aggregation

Only one reviewed article specifically listed performance measurements for non-spatial feature aggregation queries. While Aji et al. [3] discusses the need for non-spatial feature aggregation queries to be supported by a spatially aware system, only Wang and Wang [27] shows results from such queries. The results in that temporary-index Hadoop solution are shown to be on par with Oracle Spatial. While not specifically discussed, as all reviewed systems are built on a Hadoop framework, non-spatial queries can still be issued to the underlying MapReduce framework. Regarding Hadoop-GIS and SpatialHadoop, both systems have been constructed to process non-spatial queries and pass the MapReduce job to the Hadoop system. As such, it is expected that these queries would perform as any other non-spatially aware Hadoop framework with all of the benefits.

5 Implications

In this section we discuss the implications of each data organization technique in relation to the the query categories and/or individual queries. Table 1 summarizes the suitability of each architecture for each spatial query category.

As indicated in Table 1, adding persistent indexes to the Hadoop environment is beneficial to all query types, as reported in the reviewed literature [5, 12, 17, 31] (see Sect. 4 for details of reported results). Intuitively, one might expect that a non-indexed architecture that scatters nearby objects across multiple nodes would result in greater parallelism for spatial selections; however, all persistent index solutions use both physical and node based indexes, so the node based indexes are compensating for the lack of parallelism by reducing the number of computations performed.

Table 1 Suitability of architecture for spatial query categories

	Non-indexed	Temporary indexes	Persistent index
Spatial selection	No	No	Yes
Spatial join	Yes	Yes	Yes
NN variants	Yes	Insufficient	Yes
Spatial aggregation	Yes	No	Yes

The results of the Spatial Join operation indicates the join operation fits well with the MapReduce process. The implication here is that the cross-matching between large datasets is benefited by parallel processing of the Hadoop environment, regardless of the presence of spatial indexes.

Unexpected is the lack of results for Nearest Neighbor Variants. In the reviewed literature, kNN and variations are the most commonly discussed query types. Yet for temporary indexed solutions, there are not enough evaluations to provide an indication of how the architecture supports nearest neighbor queries. Both non-indexed and indexed solutions are shown to process nearest neighbor queries well.

The results here imply that the overhead in creating on-demand indexes it too significant for spatial aggregation as both non-indexed and persistent indexed solutions are shown to perform well in this area.

With regards to the results of non-indexed solutions, the special selection queries are the only category shown to not benefit from the distributed, brute-force approach of the MapReduce framework. This presents an interesting avenue for future research. A mechanism or query structure that improves the performance of spacial selection queries would mean that a standard Hadoop configuration would be a viable architecture for spatial operations in general. The indexed approaches outperform the non-indexed approaches, but require modifications to Hadoop. Furthermore, the literature contained no examples of systems with a node based index, but without a physical index. Mechanisms to construct node based indexes without requiring significant modifications to a standard Hadoop configuration could be a mechanism to achieve good performance on spatial queries in general without a highly specialized configuration.

6 Related Work

Other research has been conducted that touches upon the use of the MapReduce framework and the storage of spatial data. Zhang et al. [28] illustrates a non-indexed MapReduce solution that finds an approximation to a kNN join query. For comparison, Zhang does use an indexed Hadoop installation but the non-indexed approximation solution is the focus of the work. Using several examples, Zhang demonstrates the quality of the approximation produced by the approximate kNN join queries. As this method focuses on a technique that is not guaranteed to provide exact kNN join results, it was not included in the above comparisons.

Rather than utilize a Hadoop framework for storage and querying of spatial data, Cary et al. [10] utilizes Hadoop to generate R-tree indexes to be used by another spatial database system. The data is exported to the Hadoop system for the purpose of index creation. Those indexes are then used in a traditional DBMS that utilizes R-tree indexing on a single machine. The performance of the index creation in the Hadoop environment is used to quickly partition the data using varying data partitioning methods for evaluation purposes.

Nishimua et al. [19] describes a multi-dimensional data storage and analysis system called MD-HBase. MD-HBase is implemented as an extension to HBase. HBase [6] is similar to Hadoop using HDFS for data storage but uses a more formalized structured data format. MD-HBase uses the underlying Hadoop for the purposes of utilizing HDFS and the structure storage introduced by HBase. However, the solutions developed do not use MapReduce jobs. Instead, Nishimura compares MD-HBase to MapReduce style jobs.

7 Conclusion

In conclusion, we have discussed the various approaches taken to utilize the MapReduce framework for the purposes of storing and processing large amounts of spatial data. Various examples of spatial operations were discussed in the reviewed literature and we categorized those queries into 5 areas: Spatial Selection, Spatial Join, Nearest Neighbor Variants, Spatial Aggregation and Feature Aggregation. Each approach selected for review used the MapReduce framework but three techniques were used. Some approaches focused upon developing MapReduce algorithms using an unmodified Hadoop system. Some created indexes on-demand to aid in pruning data to expedite processing as opposed to examining every data record as is typically done in a MapReduce environment. The final approach developed persistent indexes and integrated them into the MapReduce and HDFS framework. The results show that the persistent indexes were suitable approaches for improving the performance for each of the spatial categories, and that unmodified Hadoop improves the performance for all but one category.

After reviewing the literature, it can be observed that kNN queries, or variations of the kNN query are the most commonly discussed. After this realization, it is unexpected to find an insufficient number of evaluations for this category of queries. From this observation, one need for future work is to perform more evaluations on kNN queries or queries requiring distance calculations. Additional future work is needed explore the suitability of node based indexes on an otherwise unmodified Hadoop installation, or other techniques to accelerate spatial selection queries on Hadoop.

References

1. Aji, A., Sun, X., Vo, H., Liu, Q., Lee, R., Zhang, X., Saltz, J., Wang, F.: Demonstration of Hadoop-GIS: a spatial data warehousing system over MapReduce. (2013)
2. Aji, A., Wang, F.: High performance spatial query processing for large scale scientific data. In: Proceedings of the on SIGMOD/PODS 2012 PhD Symposium, ACM, pp. 9–14. (2012)
3. Aji, A., Wang, F., Saltz, J.H.: Towards building a high performance spatial query system for large scale medical imaging data. In: Proceedings of the 20th International Conference on Advances in Geographic Information Systems, ACM, pp. 309–318. (2012)
4. Aji, A., Wang, F., Vo, H., Lee, R., Liu, Q., Zhang, X., Saltz, J.: Hadoop GIS: a high performance spatial data warehousing system over MapReduce. Proc. VLDB Endowment **6**(11), 1009–1020 (2013)
5. Akdogan, A., Demiryurek, U., Banaei-Kashani, F., Shahabi, C.: Voronoi-based geospatial query processing with MapReduce. In: IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom), IEEE, pp. 9–16 (2010)
6. Apache HBase. http://hbase.apache.org
7. Apache Hive. http://hive.apache.org
8. Blackard, J.A., Dean, D., Anderson, C.: Covertype data set. http://archive.ics.uci.edu/ml/datasets/Covertype
9. Borzsony, S., Kossmann, D., Stocker, K.: The skyline operator. In: IEEE Proceedings of 17th International Conference on Data Engineering, IEEE, pp. 421–430. (2001)
10. Cary, A., Yesha, Y., Adjouadi, M., Rishe, N.: Leveraging cloud computing in geodatabase management. In: IEEE International Conference on Granular Computing (GrC), IEEE, pp. 73–78. (2010)
11. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)
12. Eldawy, A., Li, Y., Mokbel, M.F., Janardan, R.: Cg_hadoop: Computational geometry in MapReduce. (2013)
13. Eldawy, A., Mokbel, M.F.: A demonstration of SpatialHadoop: an efficient MapReduce framework for spatial data. Proc. VLDB Endowment **6**(12), 1230–1233 (2013)
14. Ghemawat, S., Gobioff, H., Leung, S.T.: The Google file system. In: ACM SIGOPS Operating Systems Review, vol. 37, ACM, pp. 29–43 (2003)
15. Güting, R.H.: An introduction to spatial database systems. VLDB J. **3**(4), 357–399 (1994)
16. Guttman, A.: R-trees: a dynamic index structure for spatial searching. In: SIGMOD ‘84: Proceedings of the International Conference on Management of Data, ACM, pp. 47–57. New York, USA 1984
17. Liao, H., Han, J., Fang, J.: Multi-dimensional index on Hadoop distributed file system. In: IEEE Fifth International Conference on Networking, Architecture and Storage (NAS), IEEE, pp. 240–249. (2010)
18. Lu, W., Shen, Y., Chen, S., Ooi, B.C.: Efficient processing of *k* nearest neighbor joins using MapReduce. Proc. VLDB Endowment **5**(10), 1016–1027 (2012)
19. Nishimura, S., Das, S., Agrawal, D., Abbadi, A.E.: MD-Hbase: a scalable multi-dimensional data infrastructure for location aware services. In: 12th IEEE International Conference on Mobile Data Management (MDM), vol. 1, pp. 7–16. (2011)
20. Olston, C., Reed, B., Srivastava, U., Kumar, R., Tomkins, A.: Pig latin: a not-so-foreign language for data processing. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of data, ACM, pp. 1099–1110. (2008)
21. OpenStreetMap. http://www.openstreetmap.org
22. PostGIS. http://postgis.net
23. Schneider, M., Behr, T.: Topological relationships between complex spatial objects. ACM Trans. Database Syst. (TODS) **31**(1), 39–81 (2006)
24. SpatialHadoop. http://spatialhadoop.cs.umn.edu
25. TIGER Files. http://www.census.gov/geo/www/tiger/

26. Wang, K., Han, J., Tu, B., Dai, J., Zhou, W., Song, X.: Accelerating spatial data processing with MapReduce. In: IEEE 16th International Conference on Parallel and Distributed Systems (ICPADS), IEEE, pp. 229–236. (2010)
27. Wang, Y., Wang, S.: Research and implementation on spatial data storage and operation based on Hadoop platform. In: Second IITA International Conference on Geoscience and Remote Sensing (IITA-GRS), IEEE, vol. 2, pp. 275–278. (2010)
28. Zhang, C., Li, F., Jestes, J.: Efficient parallel kNN joins for large data in MapReduce. In: Proceedings of the 15th International Conference on Extending Database Technology, ACM, pp. 38–49. (2012)
29. Zhang, S., Han, J., Liu, Z., Wang, K., Feng, S.: Spatial queries evaluation with MapReduce. In: IEEE Eighth International Conference on Grid and Cooperative Computing. GCC'09, pp. 287–292. (2009)
30. Zhang, S., Han, J., Liu, Z., Wang, K., Xu, Z.: Sjmr: Parallelizing spatial join with MapReduce on clusters. In: IEEE International Conference on Cluster Computing and Workshops. CLUSTER'09, IEEE, pp. 1–8. (2009)
31. Zhong, Y., Han, J., Zhang, T., Li, Z., Fang, J., Chen, G.: Towards parallel spatial query processing for big spatial data. In: IEEE 26th International Symposium Workshops and PhD Forum of Parallel and Distributed Processing (IPDPSW), IEEE, pp. 2085–2094. (2012)

The Web KnowARR Framework: Orchestrating Computational Intelligence with Graph Databases

Edy Portmann and Patrick Kaltenrieder

Abstract This chapter presents fuzzy cognitive maps (FCM) as a vehicle for Web knowledge aggregation, representation, and reasoning. The corresponding Web KnowARR framework incorporates findings from fuzzy logic. To this end, a first emphasis is particularly on the Web KnowARR framework along with a stakeholder management use case to illustrate the framework's usefulness as a second focal point. This management form is to help projects to acceptance and assertiveness where claims for company decisions are actively involved in the management process. Stakeholder maps visually (re-) present these claims. On one hand, they resort to non-public content and on the other they resort to content that is available to the public (mostly on the Web). The Semantic Web offers opportunities not only to present public content descriptively but also to show relationships. The proposed framework can serve as the basis for the public content of stakeholder maps.

Keywords Computational intelligence · Fuzzy cognitive maps · Granular computing · Graph databases · Stakeholder management

E. Portmann (✉)
Electrical Engineering and Computer Sciences Department, Berkeley Initiative in Soft Computing, University of California, Berkeley, CA, USA
e-mail: portmann@eecs.berkeley.edu; edy.portmann@iwi.unibe.ch

E. Portmann · P. Kaltenrieder
Institute of Information Systems, Department of Information Management, University of Bern, Bern, Switzerland
e-mail: patrick.kaltenrieder@iwi.unibe.ch

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_15,

1 Introduction

Through human history, data has mostly generated and accumulated by scholars; this did not change for a long time. At the beginning of the computer age, professionals were still generating most global data, because with the invention of the computer, more and more employees were entering data into information systems.

By the end of 2002, about five Exabyte of data had been produced by mankind [9]. At the beginning of this millennium, as the Web is becoming increasingly social, users generated more of their own data (e.g., on social media platforms such as Facebook, Twitter and Wikipedia). As a consequence, data scaled from employees entering data to users entering their own data. Hence, in 2003, with the help of social media, humanity jointly generated the same five Exabyte of data again—but within a single year. This progression is accelerating [6] into big data, as now with Web 3.0 as high computational intelligence computers are generating data (e.g., Web of Things, connected sensors, and satellites [16]).

For business and commerce, the increasing volume includes both great challenges and opportunities, as big data could help companies to understand customers better and to allocate resources more effectively. As a consequence, management increasingly demands relevant data streamed to them real-time, but the data velocity and variety which is flowing into companies often exceeds the capacity of traditional information systems [11].

Combining computational intelligence techniques with suitable big data databases (e.g., graph databases [41]) allow the development of management tools to deal with increasing social Web data. To this end, we follow a design-science research approach [19], focusing on the development of a conceptual framework with the purpose of improving todays Web information systems' functionalities. The core of this framework thereby is a fuzzy ontology (i.e., unsharp-boundaries-extended traditional ontology [30, 34]) created by Web agents. Traditionally, such ontologies are a formal conceptualization of a particular domain of interest shared among heterogeneous, decentralized information systems (e.g., the Web). They entail entities, attributes, relationships, and axioms to provide a shared common-ground understanding of the world [14, 15]. Yet, there are ontological applications where information is vague and imprecise (e.g., semantic-based business and commerce Web applications). So the conceptual formalism supported by traditional ontology may not be sufficient to represent vague information. A possible solution presented in the following chapter is to incorporate fuzzy logic [53] into ontologies.

In the following chapter, we propose using self-acting agents to acquire and aggregate Web data in a nature-inspired way. We illustrate how a combination of evolutionary computing techniques with fuzzy cognitive maps (FCMs) can be harnessed to create knowledge structures of different granularity [30, 54]. This part of the chapter is based on Portmann and Pedrycz [37], who did the groundwork for the framework presented here. By combining fuzziness with neural networks, FCMs represent the first part of computational intelligence techniques, which are completed by a form of collective-intelligence-directed (i.e., automatically issued knowledge structures from users' social media interactions) evolutionary computing. With graph databases (i.e., a database that uses graph structures to represent and store big data), the emerging knowledge structures can be stored and managed as FCMs in different aggregation granularities. In this chapter, we will show step-by-step how matched Web ontologies (i.e., fuzzy ontologies aggregated, represented, and stored as FCMs) can be used as a basic framework and which can, in turn, be enriched by social Web and big data (as well as additional data) that is not directly available on the Web. Eventually, the Web agent-created fuzzy ontologies stored as FCMs (or more precisely, as FCM-graphs of different levels of granularity) are used as a base for designing a user-centered reasoning interface.

With a use case from stakeholder management, we introduce the Web KnowARR framework (the name stems from Web knower, meaning an entity that knows or apprehends particularly the World Wide Web). For the most part, the use case is taken from Portmann and Thiessen [36], but by contrast, in this chapter the focus is primarily on the processing of big data with computational intelligence techniques (i.e., applying notions of granular computing to Web data). Stakeholder management prepares a strategy utilizing information gathered during identification and analysis processes. In many cases, these processes start with stakeholder maps (i.e., visualizations of stakeholder claims and relationships [36]). So, following a design science research approach [19], an example of an innovative stakeholder mapping tool will be demonstrated. It contains search- and browsable knowledge structures (i.e., FCM-represented fuzzy ontologies) of different granularity, which can be interacted with in stakeholder management practice. For instance, to allow reasoning about big Web data, the interface consists of an enhanced stakeholder map, which (in subsequent steps) can be enriched by communication operatives with additional (by underlying agents' inaccessible) data. In the end, using FCMs of different granularity and automatized techniques will be developed to support the communication operatives in their reasoning.

The chapter is organized into five sections: The next section provides the reader with a technical background of the topic. Section 3 demonstrates the underlying Web KnowARR framework to acquire, aggregate, represent, and reason with big Web data. The forth section shows the implementation of the Web KnowARR framework in stakeholder analysis. Section 5 concludes the chapter.

2 Background

As today more and more data is generated by computers, humanity is rapidly approaching Web 3.0 [51], which is commonly characterized as a personalized Semantic Web. The Semantic Web's underlying vision can be summarized as an attempt to provide a clear description of data that can be understood (or at least processed) by computers [3]. Such a computerized use of human-intertwined Web data is only possible if information systems can independently assign meaning to natural language data. Along these lines, information may only result from Web data when the data is understood (i.e., whether by humans or computers).

A following Web 3.0 (more on the numbering in Sect. 2.1) is based on linked documents, data, and applications that are automatically generated in the course of humans' social online interactions and the Web's information that are understandable by systems. This provides an intelligent platform, where the Semantic Web (that links information) merges with social media (that connects people). By this means, as now related information overlaps with social relations, a kind of enhanced collective intelligence arises from these human interactions with the information system (cf. Portmann [34]). This collective intelligence can be used for instance for querying relevant information for stakeholder mapping. In order to introduce the Web KnowARR framework, this section illustrates the framework's underlying theories.

2.1 From Syntactic to Emergent Semantics

When users communicate through social media platforms (i.e., using natural language), they intend to convey meaning to a communication partner. The meanings they attach to particular words ride on the context these users are acquainted with, but since different people are familiar with different objects, they attach different meanings to words [10]. For humans to communicate successfully, they must be able to detect misunderstandings and correct them by a meaning negotiation process [17, 38]. For the creation of information systems with the power of communicating with humans (i.e., applying natural language), computers must be as versatile as humans in detecting and repairing misunderstandings [10, 17]. However, in comparison to humans, a computer per se has no immediate understanding of the information a sentence contains. Rather, it decomposes unstructured language into structured data that another system might understand. To do that, according to Rapaport [38], syntax may suffice for the semantics needed for natural language understanding (i.e., as desired by Web 3.0). By considering the union of the syntactic and semantic domains, semantics can be turned into a study of relations within a single domain among the markers and their interrelations.

Emergent semantics [8] builds on Rapaport's concept of syntactic semantics [38]. To capture semantics, this kind of emergence applies a closed correspondence continuum to the analysis of semantics in distributed information systems (e.g., the Web). It avails principles of learning to teach computers of (Web) information systems the meaning of data in a semiotic sense [21, 35]. Here, semiotics stands for the search for meaning (and how reality is denoted), and likewise its relevance [7, 50]. Thereby signs (e.g., words, images, objects) are not studied in isolation but as part of a semiotic sign system. By processing big data of the Web information system, it becomes feasible for a computer to automatically draw connections and create (weighted) edges [2].

Just like search agents deduce meaning (or semantics) from social interactions on the Web (i.e., from syntactic conditions), emergent semantics detects meaning from the manner in which a sign is used, as shown by Cudré-Maroux [8]. The exponential growth of data with the Semantic Web has led to an endeavor to categorize that data into facts, relationships, and entities (i.e., ontologies [14, 15]). The method applies communication and interaction in the social Web 2.0 (e.g., through social media) to automatically build semantic knowledge bases (e.g., through induction and classification [21, 35]). Hence, emergent semantics challenges this with a form of computational Darwinism (i.e., computational intelligence technique that involves continuous optimization [49]). By analyzing decentralized structures of distributed information systems, emergence is appearing through spontaneous development of new structures due to the interaction of participating entities.

Example: According to Portmann [34] the simplest form of emergent semantics is probably built from folksonomies in which Web content is tagged by users in a social, collaborative tagging process, mostly without fixed rules. Individual users assign references (so-called tags) ad libitum, resulting in long-term convergences regarding naming conventions [39], since the corresponding folksonomy reflects the authentic language and knowledge of the users. A vocabulary develops (similar to the development of a natural language), which can be signified as emergent semantic. Generally, this knowledge base can be represented with the ontology languages of the Semantic Web [18].

2.2 *Fuzzy Ontologies as Exemplification of Granular Computing*

Granular computing is a way of data processing modeled on the human ability to generalize by considering only those things that serve a specific interest [30]. Mimicking these abilities, information-systems-to-be should become able to switch among different granularities. By focusing on different granularity levels, computers should obtain different levels of knowledge, and as a result, a better understanding of the inherent knowledge structure (cf., emergent semantics). As granular computing is essential in human reasoning, it has impact on the design and implementation of intelligent systems [19, 54]. Its paradigms can be traced to rough and fuzzy sets [29, 53]. A key insight of these theories is that picking different sets of attributes commonly yields different concept granulations. A concept typically comes across as a set of entities, which are indistinguishable [25] (i.e., a concept), or a set of entities that is composed from such simple concepts (a complex concept). This can be implemented on the base of a fuzzy ontology [28, 34].

In computer science, ontologies are linguistically collected and formally ordered representations of objects and the relationships between them [14, 15]. Ontologies are used in the Semantic Web (and consequently in the Web 3.0) to exchange formal information among applications and services [18]. An emergent ontology creation method thereby differs from common Semantic Web methods. In traditional methods, Web ontologies are added top down by experts whereas in emergent methods, ontologies automatically emerge bottom up (e.g., applying natural language processing and machine learning algorithms [21, 35]). Parry [28] suggested fuzzy ontologies as a chance to overcome some of the issues associated with traditional Semantic Web ontologies.

A fuzzy ontology is rooted in the notion that each concept (consisting of an index term, object, etc.) of an ontology is related to every other concept in the ontology, with a membership degree assigned to that relationship. Stemming from Zadeh's [53] fuzzy logic (the part of computational intelligence that handles approximation), the membership degree $\mu \in [0, 1]$ denotes the strength of belonging of a concept to another concept. Fuzzy logic follows the way humans think and helps to better handle real-world facts, since human reasoning is not dichotomous, contrasting traditional logic, where all is either true or false (cf. Portmann [34]). While variables in mathematics usually take numerical results, in fuzzy logic, the non-numeric fuzzy membership relation (e.g., linguistic variables as *strongly*, *partially*, *somewhat*, etc.) are often used to cultivate the locution of rules and facts. So μ may match a linguistic variables with $\sum_{i=1}^{i=n} \mu_i = 1$ applied for each term, where $n = (N - 1)$ is the number of relations a certain object has, and N is the total number of concepts in the ontology. Note that this rule is not commutative.

Adapted from Parry [28], in Fig. 1, each μ represents the membership value of the relationship from apple to tree, fruit and computer company. Any relationships not shown are assumed to have a $\mu = 0$, and relationships directed to Apple do not have μ values shown for clarity.

In this way, the granular computing paradigm can be implemented directly. When querying the ontology, using a particular weight $\mathrm{K} \in [0, 1]$ (e.g., adjustable via a slider [34]) leads to different knowledge granules (collections of objects that are arranged together due to their similarity [25, 30, 54]). With our fuzzy ontology, knowledge structures and bases emerge self-controlled.

Many companies are storing evermore knowledge structures and knowledge bases without weighting their repositories of total human knowledge. They attempt to categorize everything without realizing that, with the intuition of building intelligent information systems [25, 30, 54], the repositories of human knowledge need somehow to mimic human use of knowledge [2]. Note that in our chapter we follow Zins' approach [56] to define data, information, and knowledge. For instance, in their everyday-reasoning, humans can weigh the relevance of a concept [50]. For computers, weighted path traversals are not new, but the processing power and memory necessary to leverage a traversal algorithm has only recently become available to business. With a fuzzy ontology, a first step towards Benedetto's suggestion [2] to focus more on relevancy [50] and less on factual accuracy is implemented. The only way to get to a Web 3.0 leads through computers performing like humans. Hence, it is important to focus on weighting edges

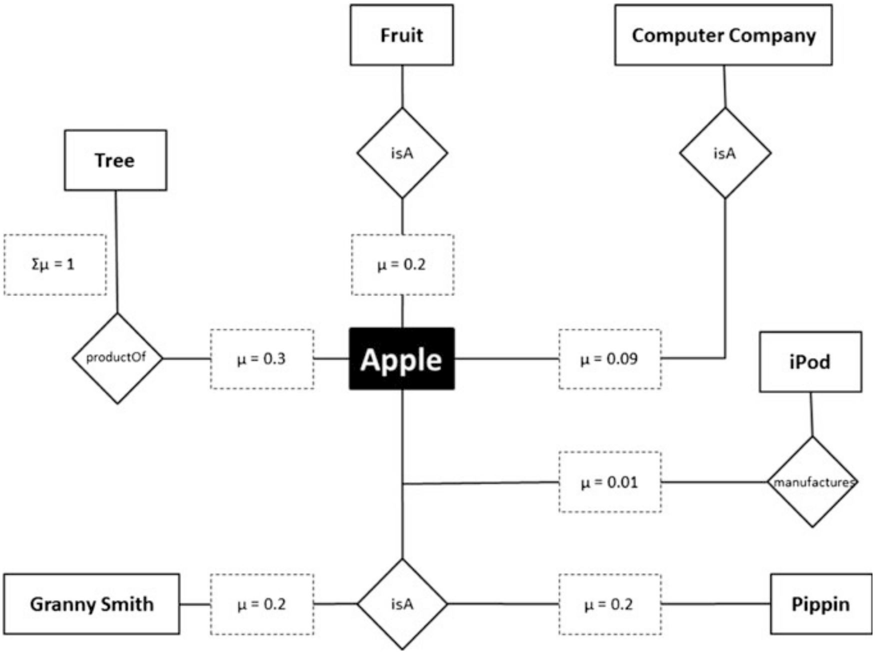


Fig. 1 A fuzzy ontology example

between knowledge structures and bases added to the Semantic Web. Fuzzy ontologies become absolutely necessary if these issues are to be solved biomimetically. A fuzzy ontology is necessary to make that relevant yet tangential connection, which is a crucial step to avoid the dreaded filter bubble [2].

2.3 Fuzzy Cognitive Maps to Manage Fuzzy Ontologies

Fuzzy cognitive maps (FCMs) that offer a viable possibility to manage fuzzy ontologies have emerged as an alternative method for representing the behavior of systems [23, 24]. As signed weighted digraphs, they usually involve feedback, consisting of nodes (concepts) and edges between those nodes [22, 42]. Fuzziness permits degrees of causality, represented as links between the concepts. Such a structure establishes forward and backward propagation of causality, allowing the knowledge base to increase when concepts and links between them are increased.

FCMs are fuzzy-neural systems [23, 43, 53] (the learning part of computational intelligence). When more data is available, the information system improves its own adaptation and reaching a solution [41]. A FCM describes the behavior of a system in terms of concepts (e.g., granular knowledge structures [52]), which is what we have outlined as fuzzy ontologies (see Fig. 1). The relationships among concepts are represented by directed links. A link thereby connects two concept models with influence of the causal variable on the effect variable. Each term used has the total membership value of its relations as a value of 1 summed over each dependent relation. For the relationship between two concepts A, B, $\mu_{AB} > \mu_{BA}$ or $\mu_{AB} < \mu_{BA}$ or $\mu_{AB} = \mu_{BA}$ are all possible.

The causal influence of the causal variable over the effect is modelled by a linking connection of both concepts. The intensity of each link is measured by its weight $w_{ij} \in [-1, 1]$, assume that the concepts are indexed by subscripts i (e.g., pre-synaptic cause node), and j (e.g., post-synaptic effect node). A matrix E stores the weights assigned to the pairs of concepts [22, 43]. Figure 2a shows metaphorically the FCM of Figs. 1 and 2b the corresponding adjacency connection matrix.

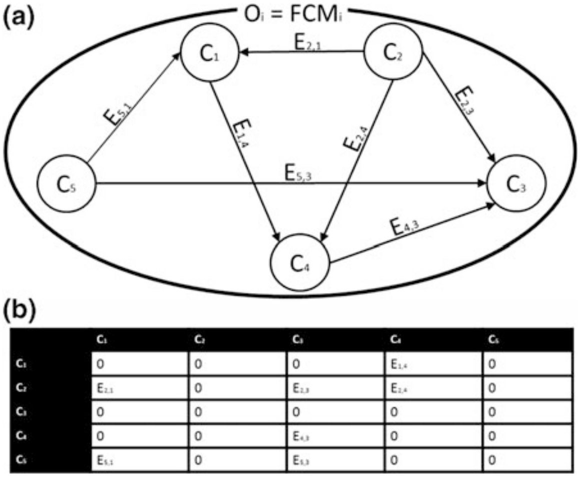
Note, for completeness, that it is possible to instantiate simple FCMs (i.e., binary maps with concept labels $A_i \in \{0, 1\}$). Their weights are typically crisp sets $e_{ij} \in \{-1, 0, 1\}$, where -1 denotes negative, 0 neutral, and 1 positive causality [22]. Each concept quantifies a degree to which the corresponding concept in the system is active at iteration step. These different kinds of possible granularities to represent FCMs reconcile ideally with fuzzy ontologies, as on a higher abstraction level one might use simple FCMs. This is possible, for example by rounding (i.e., replacing a weight for instance by another weight that is approximately equal but has a shorter and simpler representation).

2.4 Epilogue: Towards Company Stakeholder Responsibility in Web 3.0

Self-assertion of business decisions depends strongly on the integration of affected stakeholders [12]. Today, executives face decisions in a growing network of interests (groups). The complex mélange of shareholders, customers, suppliers, regulators, and socio-ethical organizations calls for the necessity to integrate claims early in the decision-making process [10]. Claims management towards company intensions is called stakeholder management. Thanks to social media, stakeholders have professionalized their campaign abilities more and more.

Today, stakeholders are able to act globally and to assert claims worldwide. Attacks of critical stakeholders are not directed only at companies anymore, but at the entire supply chain [36]. Opinion-making prospectively takes place directly, decentralized, and via social media. Claims management now goes beyond mere information activity. Methodologically, it is about creating dialogues, visualizations, moderations, and claims integrations into a decision making process [10]. Business-wise, it is about integrating claims accordingly into company processes (e.g., procurement, purchasing, corporate development, research and development).

The integration of stakeholders is value-adding relevant, because the stakeholders are able to enhance and accelerate the self-assertion of company intentions. To cover all these points, Freeman [13] introduced a concept of company-stakeholder



	C_1	C_2	C_3	C_4	C_5
C_1	0	0	0	$E_{1,4}$	0
C_2	$E_{2,1}$	0	$E_{2,3}$	$E_{2,4}$	0
C_3	0	0	0	0	0
C_4	0	0	$E_{4,3}$	0	0
C_5	$E_{5,1}$	0	$E_{5,3}$	0	0

Fig. 2 A simple FCM with corresponding **a** graph and **b** adjacency matrix

responsibility as an expansion of the traditional notion of corporate social responsibility. This new understanding is also about small and medium-sized companies' responsibilities, not only large corporations [13]. Every company thus is influenced by its stakeholders, representing all interest groups dealing with the respective company. Thereby, social responsibility is covered by stakeholder responsibility.

Analytical tools help making claims of stakeholders visible. Analogical to other management functions, they are the requirement to be able to manage stakeholder claims in the first place. In stakeholder management there exists a multitude of analytical tools. Stakeholder mapping, as one of them, thereby visualizes stakeholder claims and relationships. A part of the claims contained in these maps are not (or only hardly) publicly accessible (e.g., personal assessment, phone interviews, speech manuscripts of private events).

Yet, in contrast to five years ago, more of the claims are publicly available (e.g., on social media platforms), and thus can be used for the creation of stakeholder maps. By tapping these big data sources, now more stakeholder claims can materialize from the Web. However, without the right framework, many companies have little choice but to ignore large quantities of potentially valuable stakeholder claims. Traditional information systems are simply not optimal for acquiring, aggregating and representing the vast data, let alone automatic reasoning. With the Web KnowARR framework that we present in the following section, we take a remedial action.

3 The Web KnowARR Framework

In the Semantic Web, different actors have different interests and habits, use different tools and knowledge, and most often, at different levels of granularity. These many reasons for heterogeneity lead to diverse forms of knowledge structures and bases (which should be considered in a holistic stakeholder analysis). The connections and influences between individuals and social/organizational knowledge, goals and purpose, preferences, as well as expectations and experiences, are thus reflected in the underlying Semantic Web ontologies [18].

This section presents a framework for Web knowledge acquisition, aggregation, representation, and reasoning, where Web KnowARR is an entity that knows or apprehends the Web. Portmann and Pedrycz describe the first components of this framework [37]. Thereby, Web agents acquire and aggregate knowledge about a specific domain D. At the same time, these agents permit a wrapping W of the relevant ontologies O_i to FCMs, FCM_i. Afterwards, an alignment of different FCMs leads to the creation of a new FCM from possible overlapping submodels. In this way, the initial ontologies remain unaltered. The aligned FCM^{new} (see Fig. 3) is supposed to contain the knowledge of the respective ontologies (i.e., consequences of each ontology are consequences of the alignment). Using FCMs, we can calculate an approximated FCM. Our framework conjoins several theories and previous research (cf. [21, 34–37]) by combining various aspects and draws on the strength of these. Additionally, the use of graph databases [40] (see Sect. 3.2 et seq.) to store the Web KnowARR framework enables new ways to use computational intelligence with the vast amount of data available (i.e., big data).

Figure 3 illustrates the Web KnowARR framework that can be used for stakeholder management. Based on this framework, it is for instance possible to manage stakeholders and their actions according to Freeman's company stakeholder responsibility framework [13]. In the next subsections the three main components (*Web knowledge aggregation*, *representation*, and *reasoning*) of this framework are specified.

3.1 Knowledge Aggregation Using Fuzzy Cognitive Maps

Acquired knowledge (e.g., from big data) has first to be aggregated for further use. In the foreword to Papageorgiou's book [27], Kosko (i.e., the FCM originator) envisages a fusion of structured knowledge to an integrative FCM knowledge representation, which is addressed through this chapter. A FCMs aggregation yields thus a new FCM model FCM^{new} as a merge of submodels. The FCM underlying assumption is that combining incomplete, conflicting opinions of different Semantic Web sources may cancel out the effect of oversight, ignorance, and prejudice [1]. The simplest method is to establish the candidate FCM by averaging the corresponding relationship values (weights w) across all the submodels. Using the FCMs matrix representation (E; see Sect. 2), this is carried out, for instance, by calculating averages of the corresponding cells across all submodels.

A finite number of FCMs can be combined together to produce the joint effect of all underlying maps. Let $E_1, E_2, \ldots, E_p$ be the adjacency matrices of the maps with

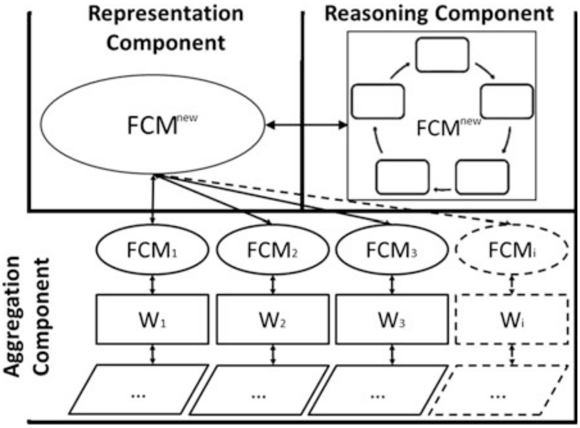


Fig. 3 The Web KnowARR Framework

$C_1, C_2, \ldots, C_n$, then the combined FCM is obtained by summing all adjacency matrices $E_1, E_2, \ldots E_p$. The combined map's adjacency matrix is labeled by $E = \sum_{i=1}^{p} E_i$, with p as the number of adjacency matrices and n as the number of concepts (see Sect. 2). Note that this standard technique can be improved by adding credibility to each submodel [48] but, for didactic reasons, we follow in this chapter the lex parsimoniae. This law suggests tending towards simpler explanations until some simplicity can be traded for increased explanatory power. Using FCMs, the next subsection illustrates the interactive visualization of the aggregated knowledge.

3.2 *Knowledge Representation Using Fuzzy Cognitive Maps*

Through dynamic query interfaces [43] humans are able to adapt to computers, and through machine learning techniques (e.g., fuzzy-neural systems) computers, in turn, can (better) adapt to humans [37]. To support communication operatives, dynamic query interfaces should fit into single users' knowledge. To do this, the computer should rely on machine learning to defer to (average) users, which use natural language. Through dynamic interfaces that integrate digital content into a human's life in seamless ways, a computer should even become adaptable and customizable to an individual user (see for example Portmann et al. [33]). With an automatically built-in fuzzy ontology, computers may become (more) responsive to humans. The aggregated FCM model (together with any underlying submodels crawled and wrapped), includes fuzzy meaning of human world knowledge (cf. Zadeh [55]) that can be used to bring together humans and computers as a part of computational intelligence [21, 35].

Let us consider concepts $C_1, C_2, \ldots, C_n$ in a simple aggregated FCM model, FCM^{new}. Suppose the graph is drawn using edge weight $e_{ij} \in \{-1, 0, 1,\}$, as for instance done with simple FCMs (see Sect. 2). The adjacency matrix $E = \left(e_{ij}\right)$, where $\mathrm{e_{ij}}$ is the weight of the directed link C_iC_j. If C_i is a concept of a FCM, then an instantaneous state vector $A = (a_1, a_2, \ldots, a_n)$, where $a_i \in \{0, 1\}$ denotes an on-off position of the concept at an instant. Hence, for $i = 1, 2, \ldots, n$:

$$a_i = \begin{cases} 0, & \text{if } a_i \text{ is off} \\ 1, & \text{if } a_i \text{ is on} \end{cases}$$

Let $\overrightarrow{C_1C_2}, \overrightarrow{C_2C_3}, \overrightarrow{C_3C_4}, \ldots, \overrightarrow{C_iC_j}$ be the links of the FCM $(i \neq j)$, then the links constitute a directed cycle. A FCM is cyclic if it possesses a directed cycle, and acyclic otherwise; a FCM with cycles is said to have feedback and is called dynamic [20]. Now, $\overrightarrow{C_1C_2}$, $\overrightarrow{C_2C_3}$, $\ldots$, $\overrightarrow{C_nC_{n-1}}$ is a cycle; if C_i is on (i.e., causality flows through the edges of the cycle) and causes C_i, the dynamic system is said to go round and round (true for any C_i, with $i = 1, 2, \ldots, n$).

However, these FCMs can be used as simple representation of knowledge, which can be represented as graphs. So, at this point, we use graph databases as storage means for the respective FCMs. Graph databases use graph structures to store aggregated FCMs of different granularities (e.g., conceivable as a kind of meta level). This form of graph database follows the intention of a modern Web-scale database [40], which allows handling of big data challenges (i.e., high volume, high velocity, and high variety [4]).

Browsing and searching as the main interactions on the Web [21, 35] allow stakeholder operatives to investigate this aggregated Web knowledge structure through an adaptive interface [46]. While browsers provide visual mechanisms for navigating the Web, in many cases search engines are the source where a Web navigation process starts. In the future, a search engine should provide a responsible communication operative the possibilities of searching and browsing not only the retrieved Web documents (keyword search), but also the underlying meaning. So, in addition to only browsing from a found Web document via link to another document, the communication operatives' search is extended by the possibility to follow a link into visualized knowledge structures (i.e., FCMs; see Fig. 2a) as a form of granular computing. Therein the communication operative navigates the netlike structures for some time, and then follows back a link to a different Web document [5].

3.3 Knowledge-Based Reasoning Using Fuzzy Cognitive Maps

The reasoning part by Portmann and Pedrycz enables conclusive logical consequences based on the fuzzy ontologies stored in the graph database as FCMs [37]. Alongside an interactive visualization of the gathered analytical content [36], these automated consequences are able to support communication operatives as a part of computational intelligence. In stakeholder management, ontology-based graphs distance themselves from less formalized structures through the fact that logical conclusions can be drawn from the knowledge stored in these graphs. As shown, via FCMs it is possible to aggregate ontologies from collective information systems. FCMs can, on one hand, be used to illustrate coherencies, and, on the other hand, they can be used in automatic reasoning (see Portmann and Pedrycz [37]).

For example, such a reasoner can ascertain equilibriums in the FCM. Equilibrium in this dynamic system is called the hidden pattern. However, we focus here more on the visualized knowledge aspects through FCMs. Compared to other systems, FCMs are relatively easy to use for representing structured knowledge, and the inference can be computed by numeric matrix operation instead of explicit if-then rules [1]. FCMs are appropriate to express the knowledge and experience which has been aggregated.

In exploring the knowledge structures, additional functionalities (e.g., zoom, drag-and-drop [34]) to standard functionalities (like clicking hyperlinks while browsing) are conceivable. Using such granular-computing-inspired functions allows communication operatives to efficiently investigate stakeholder management related knowledge, information, and data (for a comparision see also Zin's conceptual approach [56] in Sect. 2.2). So it is for example possible that with the zoom function a special part of the fuzzy ontology $O_i (= FCM_i)$ can be zoomed in [34] (i.e., subgraph analysis). After a closer investigation, a communication operative can zoom out again to the aggregated ontology $O^{new} (= FCM^{new})$. In combination with a drag-and-drop function, this allows communication operatives a knowledge-based reasoning, which, in the future, can be supported by improved, user-adapted automatic reasoning [31, 37, 47].

A simple implementation of such an automatic reasoner could be built on a query language, since graph databases are powerful tools for graph-like queries. Thereby, communication-operative-made clicks on the dynamic user interface could again serve as learning input for the reasoning system. This learned input could, in turn, be used for mathematical operations on the graph. Typical higher-level operations associated are finding paths between two nodes, finding the shortest path from one node to another, finding clusters, and/or computing diameters. It is, however, conceivable that these traditional queries can be adapted even closer to biological reasoning systems, for instance considering fuzziness of human world perceptions [10, 34]. These queries will definitively better mimic human reasoning and therefore improve underlying reasoning interfaces. An implementation could, for instance, be understood as an automatic-learned auxiliary reporting of needed higher-level graph operations. As a direct consequence, they will bring fuzzy ontology-stored computational intelligence to the user interface.

4 Deploying the Web KnowARR Framework in Stakeholder Management

Due to the rapidly increasing amount of data on the Web [4], the retrieval of analytical content has become a major challenge for companies. Often the signals are weak, data is difficult to access, output information is vague, or context cannot be seen directly [36]. Yet, company stakeholder responsibility plays an important part in today's business, as stakeholder management is essential for the success of a company. Thereby, stakeholder management basically takes place in three steps: the first step is *stakeholder analysis*, the second, *stakeholder management process* and the third, *stakeholder dialogues*.

In the stakeholder analysis, claims of individuals are compiled and visually presented as stakeholder maps. Parts of this analysis are stakeholder profiles, in which the content, as well as publications, networks, and line of argument for dialogues with the respective person are grasped. In the stakeholder management process, the drafting of a management process has priority. The question is, how stakeholders can be used as opinion leaders, which are valuable ideas providers? Finally, in the stakeholder dialogues, negotiations will be accomplished. They may range from personal conversations to interactive dialogue formats often lasting several days. The results of these dialogues are, afterwards continuously taken into account in the management process.

The presented Web KnowARR framework supports the analysis part, while the gathered Web content is becoming a part of stakeholder maps. Figure 4 illustrates a stakeholder map that is stored in the underlying graph database as a fuzzy cognitive map (FCM). For the stakeholder analysis most important subgraphs are visualized as respective stakeholder map. In the maps, personal relationships and content of opinion leaders (e.g., politicians, journalists, citizens' group leaders, project managers at non-governmental organizations) are visualized. Additional aggregated information of opinion leaders (e.g., personal attitudes, argumentations, contradictions, networks) can be accessed via their respective stakeholder profiles (by clicking the respective profile on the interactive stakeholder map interface). As a direct consequence of the vast amount of data available (i.e., big data [4]), the results of the Web analyses deliver a completely new content quality, and possibly even currently invisible relationships between opinion leaders.

In the following section we introduce the concept of company stakeholder responsibility in the age of Web 3.0 and transfer the Web KnowARR framework to stakeholder mapping. To this end, we use a case study to show how stakeholder management tools and methods can be performed with the framework.

4.1 Applying Stakeholder Maps in a Swiss Medium-Sized Company

A Swiss medium-sized company, which distributes sustainable products globally, serves as a case in point for the successful use of stakeholder maps. This use case thereby is borrowed by Portmann and Thiessen [36]. The difficulty of this company is that its direct competitors repeatedly come under fierce criticism by non-governmental organizations. The accusation is that they do not (or insufficiently) maintain sustainability standards in their respective production countries. The criticism is

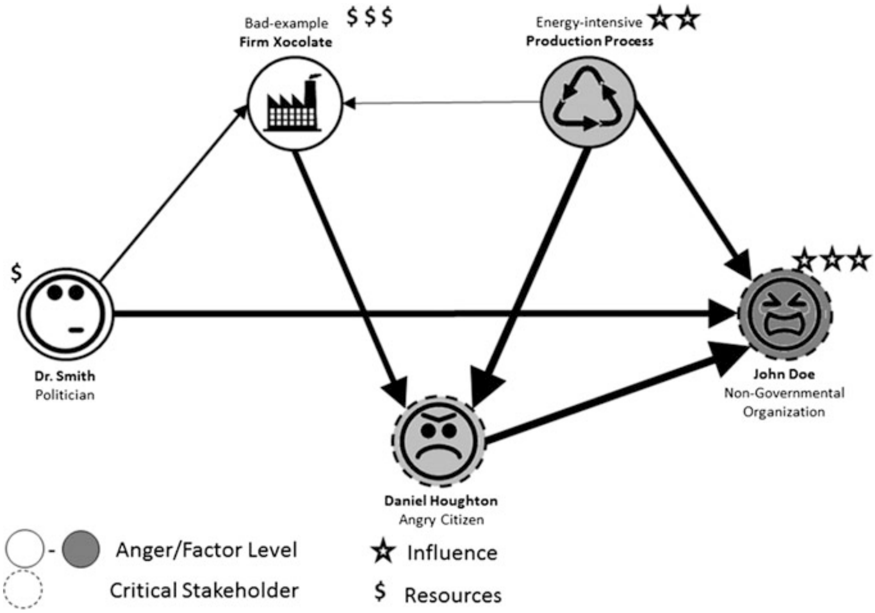


Fig. 4 Stakeholder map example

often undifferentiated and aimed generally at the entire industry. To that effect, differentiation from completing companies through its sustainable production methods is difficult for our example company.

A consequence is constant criticism by non-governmental organization concerning the company's decisions, even though the respective critiques are already considered or even consistent with the non-governmental organization's goals. The acceptance and self-assertion of company decisions is thus heavily dependent on how well the different (especially critical) stakeholders are involved in the decision-making processes [36].

The company has started to realign its management process along a company stakeholder responsibility perspective and therefore now integrates stakeholders into decisions early. The spectrum of stakeholder involvement thereby ranges from information exchange to strong collaboration (e.g., the development of common standards and good-practice guidelines). A crucial part of the newly implemented stakeholder management thus is stakeholder analysis. It aims at identifying all relevant opinion leaders for specific management tasks (visualization of opinion leaders, revealing argumentations to understand these leaders to integrate them into a long-term dialogue, etc.). Different stakeholder maps are used for such an analysis. In these maps, all opinion leaders are grouped into stakeholder groups (e.g., politicians, critical non-governmental organizations, local authorities, think tanks), and visually represented.

The critical point of such an analysis is data quality. To get a picture of relevant opinion leaders, the presented Web KnowARR framework can be applied. The opinion leaders' public contributions (e.g., political expressions, journalistic products, blogs) can systematically be identified and their arguments can be aggregated in context. The generated ontologies and their highlighted correspondences thereby help communication operatives understand stakeholders claims. To this end, the data for underlying fuzzy ontologies are on one hand aggregated from existing Semantic Web information structures (e.g., collections regarding opinion leaders), and on the other hand from Web content (e.g., blogs, social networks and/or personal webpages), as well as publicly available information in wikis and forums (e.g., about opinion leaders). The stakeholder map can hence be filled step by step with opinion leaders, their view on concrete management topics, their arguments, relevance for the decisions, and partially even their networks. In a second step, these data may then be enriched with data not available publicly and correspondingly undetectable by the Web KnowARR framework (e.g., personal networks of journalists, opinion leaders who do not publish).

4.2 Results Gained with the Web KnowARR Framework

The value of such maps is an analytical starting point to visualize opinion leaders (i.e., at individual, granular level), and to give an impression of the landscape of various opinion leaders. With the Web KnowARR framework, these maps can be automatically filled with all data publicly accessible. The great benefit of using a stakeholder management process for the respective company was that through stakeholder mapping, it was for the first time obvious what influence sustainable production has on opinion leaders, and what attitude these opinion leaders have towards the company in general. This was the basis for stakeholder profiles, in which arguments and publications were aggregated to their full extent. By understanding stakeholder arguments, joint studies and even industrial standards were developed with the advocates of sustainable production. From the beginning, critical non-government organizations were involved in the management process—and they are in the meantime welcome experts for the company. Today, executives and professionals of the company maintain their own relations with the most important opinion leaders (all of whom are respectively prepared in terms of contents, and therefore consequently contribute to the achievement of business objectives). Differentiation towards the industry has been reached, and the sweeping criticism of company decisions has almost completely ebbed. This section shows a first justification of the Web KnowARR framework based on the insights gained from applying the framework.

In the style of design science research [19], at this stage, our justification builds on Pfeifer's envisaged method [32] called understanding by building, unifying science with engineering. So, based on human-inspired fuzzy research, the emphasis is (for now) more on usability (i.e., for communication operatives) of the framework and less on exact (test) metrics. Yet, following a continuous improvement process inherent in this method (on the analogy of underlying computational Darwinism [49]), the planned implementation of our Web KnowARR framework will gradually improve step by step (i.e., following the law of parsimony; see Sect. 3.1) and, for that reason, will be tested by respective metrics later. Design science research can be roughly divided into three parts [44], namely design (i.e., creation of artifacts), public dialogue (i.e., communication of the artifact), and science (i.e., testing the artifact with metrics); so this chapter comes across more as a platform for a discussion of our tool in the works.

4.3 Working with Stakeholder Maps

The Web KnowARR framework allows an interactive examination and amendment of stakeholder maps. As a result, a tool is provided for stakeholder management, by which the increasing amount of Web data, related to the constantly more demanding monitoring and analysis of stakeholder management processes, can be managed more easily. The interactivity still permits the respective communication operatives to manually add information (i.e., claims and coherencies) to these maps.

The data stored in the fuzzy ontologies as FCM-graphs are used to automatically fill stakeholder maps with analytic content of the Web. A responsible communication operative is now able to analyze the corresponding map (i.e., with the respective search and browsing functions; see Sect. 3) and enrich it with additional claims and coherencies (with the insert function). The responsible operative has constant access to valid profile data of stakeholders at the granular level of opinion leaders. These resulting stakeholder maps can, again be saved as special graphs and hence be (further) edited later. A stakeholder analysis can and should be performed periodically in order to recognize and process changes in stakeholder attitudes over time [26]. Based on stakeholder maps (i.e., with the reasoning component of the Web KnowARR framework), it is possible to automatically draw conclusions. Based on graph theory, the reasoning component is an automated inference engine, which may deduce new statements from existing ontologies, and thus supports the communication operatives through an interactive visualization [36, 37].

Interpretation of stakeholder maps constantly takes place in relation to decisions of the company. In other words, when strategic decisions are prepared (e.g., a take-over of a company, mergers of business units, the change of a business model, controversial construction projects), it is considered from the beginning, whom this decision might relate to, and how their substantial argumentation will appear towards the decision. In order to enforce management decisions, concerned opinion leaders are involved in stakeholder management from start (e.g., through discussions, dialogues, joint elaboration of partial decisions). Stakeholder maps and underlying stakeholder profiles are constantly updated with new knowledge. They are a constant reflection of opinion leaders and an important means to help provide decision makers with acceptance and self-assertion from the beginning.

4.4 The Pros and Cons of Using the Web KnowARR Framework for Stakeholder Mapping

A big advantage of the presented Web KnowARR framework for stakeholder mapping is accuracy. Its neuroscience-based findings (cf. Shi and Griffiths [45]) in order to yield emergent ontologies [21, 35, 37], can be placed at the disposal for visualizing stakeholder maps. Additionally, because of its dependence on granular computing methods, the framework is suited to process vast amounts of data (i.e., big data). With the interaction possibilities (browsing and insert functions), communication operatives responsible for stakeholder management are able to browse, analyze and even complement information. The reasoning component supports operatives to automatically add new analytical content.

Another advantage is that the completed stakeholder map can be enriched with potential claim relationships. This allows not only working descriptively, but also to draw conclusions based on opinion and, thereby, enable much more interactive stakeholder dialogue [12, 26]. Through a specific planning of these dialogues (e.g., embedded in the management process) key issues can be identified, conflicts can be preventively anticipated, interests can be mediated, and expertise of stakeholders can be tapped [26]. An understanding of a company stakeholder responsibility may also be promoted within the company [12, 13]. In addition, stakeholder dialogues often provide opportunities for longer-term communication and collaboration with key opinion leaders, as shown by our use case.

Probably the biggest disadvantage of the self-acting Web KnowARR framework in stakeholder management can be the declining attentiveness of a responsible communication operative (i.e., through excessive confidence in the emergence of the ontology-stored FCM-graphs, and the stakeholder maps generated from it). Limitations of Semantic Web technologies are everywhere, where creative decisions or flexible problem-solving are required. An information system is (so far) only rarely able to automatically solve these tasks in a satisfactory manner, because they are mainly associated with common sense and personal preferences. Therefore, no information system will be able to take over all of these tasks completely from the person responsible (the communication operative) in the near future (i.e., using the mind and actively scrutinizing automatically generated stakeholder maps). Moreover, the framework is applicable only for publicly available data and hence may dismiss a part of the information—although this deficiency will certainly change with the implementation of the Web of Things [16].

5 Conclusion and Outlook

Technical mapping of claims and relationships in stakeholder management is still in its infancy. And yet, first tools and methods, based on the possibilities of Web 3.0, are already there, which go far beyond the current capabilities of Web monitoring and analysis. Communication understanding is simultaneously developing on the company communication side, which backs away from the pure form of information brokerage. In our contribution, we have therefore tried to bridge the gap between the technological and practical possibilities of the framework.

In order to develop mature frameworks in the future, human-centered design shall briefly be discussed as the most important requirement of Web monitoring tools. This type of design, which makes interactive tools and methods (i.e., our Web KnowARR framework adapted to stakeholder management) that consist of the highest possible usability, should be seen as guidance for future implementations of such systems. In our case, a high degree of usability (and thus also acceptance) is essentially achieved by ensuring that the respective communication operative who deals with stakeholder analysis/management is placed at the center of the implementation process with his tasks, objectives and characteristics already in place at the beginning of the implementation phase.

With this in mind, to instantiate the presented framework in a prototype, wireframes (i.e., visual guides that represents a skeletal framework of a stakeholder management cockpit) will be developed, on which the framework's usability will be examined. We will then review the wireframe with partners from practice to ensure that our requirements and objectives are met through design. In the end, to get a tangible user-centered design, we will work closely with designers and engineers to further improve our wireframe before starting implementation of a stakeholder management cockpit. According to the law of parsimony, we also plan to improve the respective algorithms in iterative processes.

In our view, it seems nothing more than logical that in development of (increasingly) automated computers, humans (i.e., the responsible communication operatives) should be at the center at any time; only they are able to assess and classify (human) stakeholder interests appropriately. These communication operatives may balance different dimensions (e.g., empathy, ethical aspects, real world phenomena, metacognitive and implicit knowledge, collective experiences and traditions, as well as simply know-how), which may be difficult to assess for computers of information systems alone. Fully automated stakeholder maps should thus be scrutinized, and if necessary be adjustable by communication operatives in an intuitive manner.

Our thesis is, however, that with (further) development of social and semantic Web technologies, computational intelligence, and the progressive professionalization of stakeholders, the relevance of company stakeholder responsibility will continue to rise. Therefore a contemporary stakeholder management will be indispensable. An important prerequisite for professionalization on the company side is technical systems that leave traditional thinking patterns and provide possibilities to support the decision-making of companies. Through the presented Web KnowARR framework, the first step in the right direction is taken, on which a stakeholder management cockpit can be based that supports communication operatives in stakeholder management.

References

1. Aguilar, J.: A survey about fuzzy cognitive maps papers. Int. J. Comput. Cogn. **3**(2), 27–33 (2005)
2. Benedetto, J.: Let's build a semantic web by creating a Wikipedia for relevancy. http://gigaom.com/2013/11/24/lets-build-a-semantic-web-by-creating-a-wikipedia-for-relevancy/ (2013)
3. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Sci. Am. J. **284**(5), 28–37 (2001)
4. Beyer, M.: Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data. In: Gartner Group (2011)
5. Bizer, C., Heath, T., Berners-Lee, T.: Linked data—the story so far. Int. J. Semant. Web Inf. Syst. **5**(3), 1–22 (2009)
6. Borne, K.: Collaborative annotation for scientific data discovery and reuse. Bull. Am. Soc. Inf. Sci. Technol. **39**(4), 44–45 (2013)
7. Chandler, D.: Semiotics the Basics. Routlege, London (2007)
8. Cudré-Mauroux, P.: Emergent semantics. In: Ling, L., Tamer Ozsu, M. (eds.) Encyclopedia of Database Systems. In: Springer, Berlin, 982−985 (2009)
9. Dimandis, P.H., Kotler, S.: Abundance: the Future Is Better than You Think. Free Press, New York (2012)
10. Dimitrov, V., Russell, D.: The Fuzziness of Communication. In: Fell, L., Russell, D., Stewart, A. (eds.) Seized by Agreement, Swamped by Understanding. http://www.pnc.com.au/∼lfell/fuzcom.pdf (1994)
11. Franks, B.: Taming the Big Data Tidal Wave Finding Opportunities in Huge Data Streams with Advanced Analytics. Wiley Hoboken, Ney Jersey (2012)
12. Freeman, R.E.: Strategic Management a Stakeholder Approach. Cambridge University Press, Cambridge (1984)
13. Freeman, R.E., Velamuri, S. R., Moriarty, B.: Company stakeholder responsibility: a new approach to CSR. Business Roundtable, Institute for Corporate Ethics, Bridge Paper. http://www.corporate-ethics.org/publications/bridge-papers/ (2006)
14. Gruber, T.: A translation approach to portable ontology specifications. Knowl. Acquisition **5**(2), 199–220 (1993)
15. Gruber, T.: Collective Knowledge Systems: Where the Social Web meets the Semantic Web. J. Web Semant. **6**(1), 4–13 (2008)
16. Guinard, D., Trifa, V.: Towards the Web of Things: Web Mashups for Embedded Devices, WWW2009, April 20–24, Madrid, Spain (2009)
17. Hirst, G.: Negotiation, compromise, and collaboration in interpersonal and human-computer conversations, In: AAAI Technical Report WS-02-09 (2002)
18. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. CRC Press, Boca Raton (2010)
19. Johannesson, P., Perjons, E.: A Design Science Primer. In: Create Space Publisher (2012)
20. Kandasamy, W.B.V., Samarandache, F.: Fuzzy Cognitive Maps and Neutrosophic Cognitive Maps. Phoenix, Xiquan (2003)
21. Kaufmann, M.A., Portmann, E., Fathi, M.: A Concept of Semantics Extraction from Web Data by Induction of Fuzzy Ontologies. In: IEEE International Conference on Electro/Information Technology, Rapid City, SD, USA (2013)
22. Kontogianni, A.E., Papageorgiou, E.I., Tourkolias, C.: How do you perceive environmental change? Fuzzy Cognitive Mapping informing stakeholder analysis for environmental policy making and non-market valuation. Appl. Soft. Comput. **12**, 3725–3735 (2012)
23. Kosko, B.: Fuzzy cognitive maps. Int. J. Man Mach. **24**, 65–75 (1986)
24. Kosko, B.: Neural Networks and Fuzzy Systems. Prentice-Hall, Englewood Cliffs (1992)
25. Lin, T.Y.: Granular computing: fuzzy logic and rough sets. Computing with Words in Information/Intelligent Systems 1. Physica-Verlag HD, 183–200 (1999)

26. Lintemeier, K., Thiessen, A., Rademacher, L.: Stakeholder Integration: Zum Wertschöpfungsbeitrag von Unternehmenskommunikation und Nachhaltigkeitsmanagement. Steinhausen, München (2013)
27. Papageorgiou, E.I.: Fuzzy Cognitive Maps for Applied Sciences and Engineering: From Fundamentals to Extensions and Learning Algorithms. Intelligent Systems Reference Library 54. Springer, Heidelberg (2014)
28. Parry, D. T.: Fuzzy ontology and intelligent systems for discovery of useful medical information. PhD Thesis, Auckland University of Technology (2005)
29. Pawlak, Z.: Rough sets. Int. J. Parallel Prog. **11**(5), 341–356 (1982)
30. Pedrycz, W.: Granular Computing: Analysis and Design of Intelligent Systems. CRC Press, Boca Raton (2013)
31. Pezulo, G., Calvi, G., Castelfranchi, C.: DiPRA: Distributed Practical Reasoning Architecture. In: International Joint Conference on Artificial Intelligence, pp. 1458–1463 (2007)
32. Pfeifer, R., Scheier, Ch., Riegler, A.: Understanding Intelligence. MIT Press, Massachusetts (2001)
33. Portmann, E., Andrushevich, A., Kistler, R., Klapproth, A.: Prometheus—Fuzzy Information Retrieval for Semantic Homes and Environments. In: Proceeding for the third International Conference on Human System Interaction, Rzeszów, pp. 757–762 (2010)
34. Portmann, E.: The FORA Framework—a Fuzzy Grassroots Ontology for Online Reputation Management. UniPrint, Fribourg (2012)
35. Portmann, E., Kaufmann, M.A., Graf, C.: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, Maui, Hawaii, USA (2012)
36. Portmann, E., Thiessen, A.: Web 3.0 Monitoring im stakeholder management. In: Andreas Meier and Marcel Blattner (eds.) Web Monitoring, HMD edn 293, vol. 50. Jahrgang. dpunkt.verlag GmbH, Heidelberg (2013)
37. Portmann, E., Pedrycz, W.: Fuzzy web knowledge aggregation, representation, and reasoning for online privacy and reputation management. In: Elpiniki Papapgeorgiou (ed.) Fuzzy Cognitive Maps for Applied Sciences and Engineering - From Fundamentals to Extensions and Learning Algorithms, Intelligent Systems Reference Library. Springer (2014)
38. Rapaport, W.J.: What Did You Mean By That? Misunderstanding, Negotiation and Syntactic Semantics. J. Mind Mach. **13**, 397–427 (2003)
39. Rebstock, M., Fengel, J., Paulheim, H.: Ontologies-Based Business Integration. Springer, Berlin (2008)
40. Robinson, I., Weber, J., Eifrém, E.: Graph Databases. O'Reilly Media, Sebastapol (2013)
41. Rodriguez-Repiso, L., Setchi, R., Salmeron, J.L.: Modelling IT projects success with fuzzy cognitive maps. Expert Syst. Appl. **32**(2), 543–559 (2007)
42. Salmeron, J.L.: Modelling grey uncertainty with fuzzy grey cognitive maps. Expert Syst. Appl. **37**(12), 7581–7588 (2010)
43. Salmeron, J.L.: Fuzzy cognitive maps for artificial emotions forecasting. Appl. Soft Comput. **12**(2), 3704–3710 (2012)
44. Schunn, Ch. D.: How kids learn engineering: the cognitive science perspective. Bridge Linking Eng. Soc. **39**(3), 32–37 (2009)
45. Shi, L., Griffiths, T.L.: Neural implementation of hierarchical Bayesian inference by importance sampling. In: Proceedings of Advances in Neural Information Processing Systems, pp. 1669–1677 (2009)
46. Shneiderman, B., Plaisant, C.: Designing the User Interface, 4th edn. Person/Addison-Wesley, Boston (2005)
47. Simou, N., Kollias, S.: Fire: A fuzzy reasoning engine for imprecise knowledge. Berlin (2007)
48. Stach, W., Kurgan, L., Pedrycz, W.: A divide and conquer method for learning large Fuzzy Cognitive Maps. Fuzzy Sets Syst. **161**, 2515–2532 (2010)
49. Valiant, L.: Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World New York. Basic Books (2013)

50. Wilson, D., Sperber, D.: Meaning and Relevance. Cambridge Press, New York (2012)
51. Wolfram, C.: Communicating with apps in web 3.0 IT PRO, 17 Mar 2010 (2010)
52. Xirogiannis, G., Glykas, M.: Fuzzy cognitive maps in business analysis and performance driven change. IEEE Trans. Eng. Manage. **51**(3), 334–351 (2004)
53. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**(3), 338–353 (1965)
54. Zadeh, L.A.: Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. Soft. Comput. **2**, 23–25 (1999)
55. Zadeh, L.A.: A note on web intelligence, world knowledge and fuzzy logic. Data Knowl. Eng. **50**, 291–304 (2004)
56. Zins, Ch.: Conceptual approaches for defining data, information, and knowledge. J. Am. Soc. Inform. Sci. Technol. **58**(4), 479–493 (2007)

Part III
Case Studies

Customer Relationship Management and Big Data Mining

Yi Hui Liang

Abstract Successful customer relationship management (CRM) requires enterprises to interact flexibly with their customers. Enterprises must quickly and effectively find complex customer data from large quantities of data by big data mining to help understand and interact with them by suitable marketing tactics, increase the value to the customer, and improve their competitive advantages of enterprises. In this chapter, discuss big data mining, customer relationship management, customer value, and propose a case study of big data mining for customer relationship management with data of the Automotive Maintenance Industry.

Keywords Big data mining · Customer relationship management · Customer value

The chapter mainly discusses customer relationship management and big data mining. This chapter is structured as the following. Section 1 discusses big data mining. Section 2 discusses customer relationship management. Section 3 discusses the relationship between big data mining and customer relationship management. In Sect. 4, presents a case study of big data mining for customer relationship management with data of the Automotive Maintenance Industry.

1 Big Data and Big Data Mining

With the fast development of networking, data storage, and the data collection capacity, the world has experienced a dramatic increase in our capabilities to collect data from different devices, in various formats, from independent or related applications. *Big Data* is a new term used to recognize the datasets that due to their

Y. H. Liang (✉)
Information Management Department, I-SHOU University, Kaohsiung 84001, Taiwan, ROC
e-mail: german@isu.edu.tw

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_16,

vast size and intricacy, and cannot been managed them with our present methodologies or data mining software instruments. The origin of the term *Big Data* is owing to the fact that we are making a large amount of data each day. Big Data involve vast volume, complex, growing data sets with multiple, autonomous sources. Big data is so big as to be difficult to work with using most relational database management systems and desktop statistics and visualization packages, and process using on-hand database management tools or traditional data processing applications. What is considered big data differs relying on the capabilities of the organization managing the data set, and on the capabilities of the applications that are applied to process and analyze the data set in its realm. Big data are now fast spreading in all science and engineering areas. The Big data challenge is becoming one of the most heart-stirring opportunities for the future years. Data mining is one of these opportunities.

Data mining (the analysis step of Knowledge Discovery and Data Mining process; KDD) indicate that a process of extraction of implicit, prior unknown and potentially helpful [4]. Data mining technology can be employed to excavate out hidden information behind the data and identify useful patterns and associations [19]. Data mining is gradually more and more important for businesses. Big Data mining was appeared from the beginning, as the first book mentioning *Big Data* is a data mining book that appeared also in 1998 by Weiss [27]. However, the first academic study with the words *Big Data* in the title appeared a little later in 2000 in a study proposed by Diebold) [7]. Big data mining is the capability of extracting useful information from these large datasets or streams of data, that due to its volume, variability, and velocity, it was not possible before to do it [9]. Big data mining is the expansive form of data mining. The difference between Big Data mining and Data mining is that big data mining is especially emphasized for *Big Data*.

2 Customer Relationship Management

Customer relationship management (CRM) denotes that managerial efforts to business processes and technologies that designed to understand the customers of a firm [15]. Successful CRM requires enterprises to interact flexibly with their customers [8]. Enterprises that succeed in correctly assessing customer value can offer customized services to diverse customers, perform effective customer relationship management and, simultaneously, also increase enterprise revenues [1]. Kotler [17] defined customer value as the difference between the benefits which enterprises obtained from customer of enterprises and the costs incurred in attracting and serving customers. Customer value is fundamental to customer relationship management. It can be a starting point of customer relationship management to understand and measure the true value of a customer [15]. As enterprises successfully improve the lifetime value of customers, they will improve their investment returns, enhance customer loyalty, increase the profits from the existing customers, and improve the value of their database, and so on.

Kahan [14] supposed that RFM rule is easy to use and implemented quickly. RFM rule is generally acknowledged as the most popular customer value analytical method at present. McCarty and Hastak [20] suggested that RFM rule is an acceptable procedure in any circumstances, except for low response rate to a mailing and mailing relatively small portion of database in direct marketing.

RFM rule is used to analyze customer value by enterprises with three dimensions. The three dimensions are (recency; the time recently), M (money; the total amount of money) and F (frequency; the frequency).

The R variable means the difference between the time that customer purchased the last time and the time that analyze now.

The M variable means the total amount of money of enterprise product which customers bought during certain time.

The F variable means the frequency of the products customer bought during certain time (in one month, one season or one year).

3 Customer Relationship Management and Big Data Mining

Enterprises can quickly and effectively find complex customer data from large quantities of data by data mining to help understand and interact with them by suitable marketing tactics, increase the value to the customer, and improve their competitive advantages of enterprises [3]. Cheng et al. [6] supposed that data mining technology could help business in customer relationship management as listed in the following:

1. Improve business efficiency in the least budget.
2. Utilize database marketing to maintain customer relationship.
3. Increase customer loyalty and customer value contribution, decrease customer loss rate.
4. Learn customer need to develop strategy.
5. Evaluate the effectiveness of advertisement and promotion.
6. Control competitive advantages and improve brand orientation.
7. Respond to the expectation of customer and strengthen service quality.

The main function of big data mining technology for customer relation management is divided into five kinds. The five kinds are classification, estimate, clustering, association rule and prediction [6].

Clustering is generally utilized for market segmentation. Market segmentation subdivides the customers into distinct subsets of customers, where any subsets may conceivably be chosen as a target market to be reached with a distinct marketing mix [17]. Smith [25] proposed the concept of market segmentation and indicated that market segmentation involves viewing a heterogeneous market as many smaller homogenous markets, in response of differing preferences, attributable to the desires of customers for more accurate satisfactions of their varying wants. Kotler [17] supposed that market segmentation can be used to identify and profile distinct groups of buyers who might prefer or require varying products and marketing mixes and next these enterprises decides which segments present the greatest opportunity. Until now, above concept still play a crucial role.

There are two main categories of segmentation methods using big data mining technologies—multivariate statistical analysis and the neural network model.

For multivariate statistical analysis, K-means approach belongs to one kind of multivariate statistical analysis. K-means method is especially suitable when the number of observations is more or the data file is enormous [28]. K-means method is widely used to market segmentation [11, 13, 15, 24].

Neural networks can be divided for supervised, unsupervised, associate and optimization application network. SOM approach [16] belongs to one kind of the unsupervised neural network model. SOM network is usually used to market segmentation [2, 12, 18, 23].

Decision trees is an important technique in big data mining, are used extensively in classification. The advantages of decision tree theory include (1) can produce understandable rules; (2) perform tasks without much computing; (3) can handle continuous and categorical variables; (4) can learn which attributes are important for classification [5]. The disadvantages of decision tree theory are (1) presumes that the input data are relational; (2) every training data set is presumed to pertain to a predefined class ascertained by some of the attributes [5].

This section discusses the relationship between customer relationship management and big data mining, and demonstrates the available big data mining methods for customer relationship management. The methods such as multivariate statistical analysis and the neural network model. Otherwise, granular computing involves a host of new concept and techniques. Specially deserving note are techniques which relate to classification [22]. The above granular computing also can be used in customer relationship management for market segmentation.

4 Case Study: Adopting Big Data Mining Technologies for Customer Relationship Management with Data of the Automotive Maintenance Industry

In this section, propose a case study of big data mining for customer relationship management with data of the Automotive Maintenance Industry in Taiwan [19]. The Commercial Trade Council of the Automotive Maintenance Industry in Taipei estimated the number of the domestic automotive maintenance factories in Apr 2007 as nearly 9,180, of which 300 were registered as the first class, 700 were registered the second class, while nearly 8,000 were smaller ones. However, many factories have not registered according to the law, and the estimated number of factories is three times the number registered according to regulation.

The handout issued by the Bureau of Employment and Vocational Training in Taiwan states that the automobile maintenance industry has three characteristics:

1. It is service-driven.
2. It highly emphasizes customer satisfaction.

It must constantly strengthen customer management. Also there are a big data in their storage.

Many studies have employed big data mining to analyze customer data until now, but few studies have tried to systematically integrate numerous data mining technologies within the same field.

Therefore, this case study analyzes customer data for Taiwanese automotive maintenance industry by systematically integrating data mining approaches to analyze customer value. First, the K-means and SOM methods are adopted to perform customer value analysis and segment customers based on customer value. Secondly, decision tree is used to mine the characteristics of each customer segment. Third, different strategies are developed for differently valued customer segments and customer value is thus promoted.

Research method of this study is described as follows:

A. Transfer the data into R, M and F variables

This research at first transferred the data of sample data set into R, M and F variables, as the input variables of K-means method and SOM network.

B. The K-means and SOM methods are adopted to perform customer value analysis and to segment customers

Secondarily, use K-means method and SOM network to build model to analyze R, M and F variables in order to carry on customer value analysis, find out high, middle and low value of customers and compare the results of two methods. The analytical steps of K-means method are as follows [21]:

Step 1. Initialize:

Choose the number of cluster, k. For each of these k clusters chooses an initial cluster center: $\{c_1(m), c_2(m), \ldots, c_k(m)\}$

where $c_j(m)$ represents the value of the cluster center at the mth iteration.

Step 2. Distribute samples:

Distribute all sample vectors.

$$x_p \in \vartheta_j(m)$$

If $\left\|x_p - c_j(m)\right\| < \left\|x_p - c_i(m)\right\|$ for all

$$i = 1, 2, \ldots, k,\ i \neq j,$$

$\vartheta_j(m)$ represents the population of cluster j at iteration m.

Step 3. Calculate new cluster centers:

$$c_j(m+1) = \frac{1}{M} \sum_{x_p \in \vartheta_j(m)} x_p$$

where M_j is the number of sample vectors attached to ϑ_j during Step 2.

Step 4. Check for convergence:

The condition for convergence is that No cluster center has changed its position during Step 3.

Second, the analytical steps of SOM network are as follows [10]:

Step 1. Initialize weights w_{ij}:

Set topological neighborhood parameters. Set learning rate parameters.

Step 2. While stopping condition is false, do Step 3–9.
Step 3. For each input vector x, do Step 4–6.
Step 4. For each j, compute:

$$D(j) = \sum_i (w_{ij} - x_i)^2.$$

Step 5. For index J such that $D(j)$ is a minimum.
Step 6. For all units j within a specified neighborhood of J, and for all i: $w_{ij}^{new} = w_{ij}^{old} + \alpha(x_i - w_{ij}^{old})$.
Step 7. Update learning rate.
Step 8. Reduce radius of topological neighborhood at specified times.
Step 9. Test stopping condition.

C. Utilize decision tree to mine the characteristics of every customer segment

Third, use decision tree theory to build model in order to get the decision rules based the analyzed results using K-means method or SOM network. The analytical steps of decision tree are as follows [5, 26]:

Step 1. Initialize:

Use S represents the analyzed results using K-means method or SOM network.

$$S = \{C_1, \mathrm{C}_2, \ldots, C_i, \ldots, C_n\}$$

C_i represents the ith cluster.

Step 2. If $C_i, i = \{1, 2, 3, \ldots, n\}$ in the training data.

freq(C_i, S) represent the number of data in every class.
$|S|$ represent the number of data in the all training data.

Therefore, $freq(C_i, S)/_{|S|}$ represents the ratio of the *i*th cluster in the training data. Based the decision tree theory, the information of *i*th cluster is $-\log_2^{\{freq(C_i, S)/|S|\}}$.

The entropy *info(S)* of *i*th cluster is $-\sum_{i=1}^{n} \frac{(\mathrm{C_i,S})}{|S|} \log_2^{\{freq(C_i, S)/|S|\}}$.

Step 3. Divide S into m subsets $(S_1, S_2, S_3, \ldots, S_m)$ according to attribute A.

The entropy after the division is:

$$\mathrm{info}_A(S) = -\sum_{i=1}^{n} \frac{|S_i|}{|S|} \times \mathrm{info}\ r(S_i).$$

The information gained owing the division based on attribute A represent as:

$$gain(A) = \mathrm{info}r(S) - \mathrm{info}r_A(S)$$

Step 4. Select the largest value as the segmenting attribute among the $gain(A)$ value of every attribute which is calculated. Continue the above cycles and stop until reach no segmentation.

This case study systematically integrates three data mining technologies to analyze customer value.

D. Developed different strategies for customers with different values

Validation of experimental results of this case study is described as follows:

A. Transfer the data into R, M and F variables

This case study chose the factory of automotive maintenance industry in southern Taiwan as the sample, collected sample customer data and set up the 16 months database for the transaction history of customer (from the beginning January 1993 to the end April 1994). List every field variable of the database as follows in Table 1:

B. The K-means and SOM methods are adopted to perform customer value analysis and to segment customers

This research picks and fetches R, F, M variables as the input variables of K-means method and SOM network. The analytical result based the analytical steps of K-means method are as follows:

F value of 3 clusters and 4 clusters is more significantly than 2 clusters. F value of 3 clusters and 4 clusters is similar, but the number of cluster 1in 4 clusters is 9. So, adopt 3 clusters in K-means method.

The analytical result based the analytical steps of SOM method are as follows:

F value of 3 clusters and 4 clusters also is more significantly than 2 clusters. F value of 3 clusters and 4 clusters is similar, but average R, M and F variables in 3

Table 1 The field variable of the database

	Field variable	Remarks
1	The number of work form	Serial number figure
2	The date of entering factory	Express with the Christian era date
3	The name of car owner	Chinese
4	Car style	English or Chinese
5	Car plate	English + figure
6	The number of mileage	Figure
7	The committed item	Divided into a part and changed with the salary
8	Wage	Figure
9	Part name	More than 500 kinds of part names
10	Quantity	Number that the part changes
11	Unit price	Part price per unit
12	Amount of money	The salary and price of part are added together

clusters are more distinct than those in 4 clusters in magnitude order. Therefore, the study adopts 3 clusters in SOM method. Compare the result of K-means method with that of SOM method in Table 2. Average R, M and F variables all reveal clear and identical relation in K-means and SOM method.

As shown in Table 3, the F value and P value of K-means method are larger than those of SOM method. Consequently, this study used K-means method in analyzing customer value.

This study suggested the 3 clusters results of K-means method, and carry on further analysis using LSD and Turkey HSD tests to test whether one cluster is significantly different with the other group in R, M and F variables. LSD and Turkey HSD tests all show the same result. Cluster 1 and cluster 2 are not significantly different at the 0.05 level, but cluster 3 and cluster 1, cluster 3 and cluster 2 both are significantly different at the 0.05 level for R variable. Cluster 1 and cluster 2, cluster 3 and cluster 1, cluster 3 and cluster 2 all are significantly different at the 0.05 level for M variable. Cluster 1 and cluster 2, cluster 3 and cluster 1, cluster 3 and cluster 2 all are significantly different at the 0.05 level for F variable. M and F variables are better than R variable in clustering through LSD and Turkey HSD tests.

This case study adopted 3 clusters of K-means method and define cluster 1 as high value customer, cluster 2 as medium value customer and cluster 3 as low value customer in Table 4. High value customer occupied 7.37 %, medium value customer occupied 25.67 % and low value customer occupied 66.96 % of all customers.

C. Utilize decision tree to mine the characteristics of every customer segment

This chapter utilizes decision tree theory to mining the characteristics of each segment. This database collects a total of 448 number of car, take 373 number of cars as training data and 75 number of car as testing data according to that the proportion is 5:1. Figure 1 shows the analytical result. There are three rules in total. The reliabilities of all rules are acceptable because the values of reliability

Table 2 The result for the number of cluster = 3

	Cluster	Average R	Average M	Average F
K-means	1	118	58,225	6
	2	146	20,053	3
	3	209	4,444	1
SOM	1	112	57,655	6.31
	2	155	20,091	3.23
	3	232	5,656	1.53

Table 3 The result for the number of cluster = 3

	F value	*P* value
K-means	453.649	0.000
SOM	91.501	0.000

are fallen between 0 and 1. Figures 2 and 3 illustrate that the error rate is 0 and the training data set and testing data set entirely match these rules. Based on three rules, the trading money are larger than 39,100 belong to high value customer; between 12,200 and 39,100 belong to medium value customer; below 12,200 belong to low value customer.

This case study adopted 3 clusters of K-means method and define cluster 1 as high value customer, cluster 2 as medium value customer and cluster 3 as low value customer in Table 4. High value customer occupied 7.37 %, medium value customer occupied 25.67 % and low value customer occupied 66.96 % of all customers.

D. Developed different strategies for customers with different values

This research interviewed the sample factory above. The interview responses revealed that the factory only advised customers that mileage of car was begin to expire and needing repair and maintenance in the strategy as part of its customer relationship management strategy. Moreover, the factory treated all customers with the same strategy. It did not develop different strategies for customers with different values and thus resulting in cost-wastage and inefficiency. We proposes promoting customer value strategies based on the above analytical results. The four strategies are mileage advice, consumption cards, membership cards, and special projects. Mileage advice indicates that advising customers that their mileage is about to expire and remind customers to repair and maintain their vehicles. Consumption cards indicates that utilizing consumption cards to offer certain discounts and raise customer loyalty. Membership cards indicates that joining customers to become the members of factory for automotive maintenance industry, distribute membership cards and offer favors to members. It can also improve customer loyalty. Special projects indicates that providing products at special prices and other activities can help customers to maintain and enhance customer loyalty.

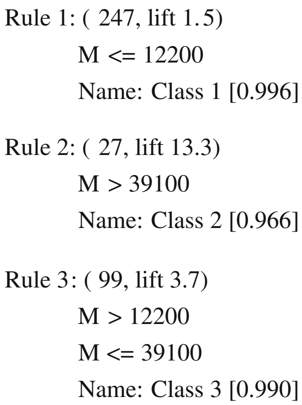
Make the following suggestions for different customer groups in the section:

1. High value customers: For high value customers, their spending the amount of money is the most highest among customers. Therefore, efforts should be made to attract high value customers and promote their loyalty by providing

Table 4 The result for customer value

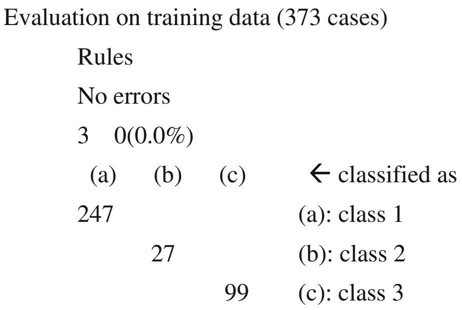
Customer Value	Average R	Average M	Average F	The number
High value	118	58,255	6	33
Medium value	146	20,053	3	115
Low value	209	4,444	1	300

Fig. 1 The analytical result of decision tree theory



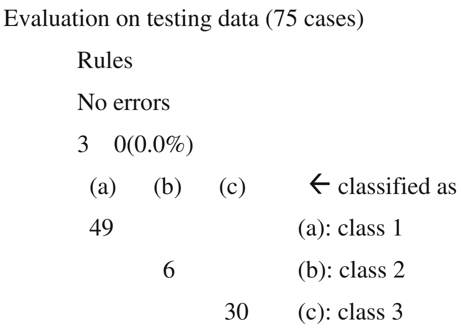
```
Rule 1: ( 247, lift 1.5)
        M <= 12200
        Name: Class 1 [0.996]

Rule 2: ( 27, lift 13.3)
        M > 39100
        Name: Class 2 [0.966]

Rule 3: ( 99, lift 3.7)
        M > 12200
        M <= 39100
        Name: Class 3 [0.990]
```

Fig. 2 The analytical result of the training data



```
Evaluation on training data (373 cases)
        Rules
        No errors
        3   0(0.0%)
         (a)    (b)    (c)         ← classified as
        247                        (a): class 1
                27                 (b): class 2
                       99          (c): class 3
```

Fig. 3 The analytical result of the testing data



```
Evaluation on testing data (75 cases)
        Rules
        No errors
        3   0(0.0%)
         (a)    (b)    (c)         ← classified as
         49                        (a): class 1
                 6                 (b): class 2
                       30          (c): class 3
```

for them memberships of factory for automotive maintenance industry, offering special services, improving existing feedback projects regularly, and increasing the additional value of membership cards, in order to promote loyalty of high value customer.

2. Medium value customers: Medium value customers comprised approximately 34.96 % of turnover in this sample. Therefore, this study suggested that businesses adopt mileage advice, consumption cards and special projects encouraging customers to consume by this group of customers.
3. Low value customers: This group of customers had an average transaction frequency of just once during the study period (16 months), and average total spending of just 4444 NTD during 16 months, and average recency on traction was 209 days during 16 months and this implicated average period of more half a year since their most recent transaction, indicating that they were a low value group. The spending of low value customers approximately only accounted for just 6.75 % of turnover. This group comprised dissociated customers. This study suggested that businesses merely provide mileage advice to this group. Businesses thus adopted a conservative approach, focusing management and service resources primarily on high and medium value customers. Despite the proposal of adopting a conservative approach to low value customers, involving a reduction of service quality and no provision of the additional services similar to those offered to high and medium value customers, by adopting a conservative approach to or a reduced level of service to low value customers, businesses can shift the resources saved to high and medium value customers.

5 Conclusions

Big data mining is now widely used in all fields because of its purposeful usage in the field where it is applied to. In this chapter, mainly investigates customer relationship management and big data mining, and present a case study of big data mining for customer relationship management with data of the Automotive Maintenance Industry. The case study is given which emphasized the used of the big data mining techniques in customer relationship management.

References

1. Berson, A., Smith, S., Smith, M., Thearling, K.: Building Data Mining Applications for CRM. McGraw-Hill, New York (2000)
2. Bloom, J.Z.: Tourist market segmentation with linear and non-linear techniques. Tour. Manag. **25**(6), 723–733 (2004)
3. Cerny, P.A.: Data mining and neural networks from a commercial perspective. In: the 36th annual ORSNZ conference. Christchurch, NZ (2001)

4. Chen, M.S., Han, J., Yu, P.S.: Data mining: an overview from a database perspective. IEEE Trans. Knowl. Data Eng. **8**(6), 866–883 (1996)
5. Chen, Y.L., Hsu, C.L., Chou, D.C.: Constructing a multi-valued and multi-labeled decision tree. Expert Syst. Appl. **25**(2), 199–209 (2003)
6. Cheng, B.W., Chang, C.L., Liu, I.S.: Enhancing care services quality of nursing homes using data mining. Total Qual. Manag. **16**(5), 575–596 (2005)
7. Diebold, F.X.: 'Big Data' dynamic factor models for macroeconomic measurement and forecasting. In: Dewatripont, M., Hansen, L.P., Turnovsky, S. (eds.) Advances in Economics and Econometrics: Theory and Applications, Eighth World Congress of the Econometric Society. Cambridge University Press, Cambridge, pp. 115–122 (2003)
8. Edelstein, H.: Building Profitable Customer Relationships with Data Mining, Executive Briefing. SPSS inc., Chicago (2000)
9. Fan, W., Bifet, A.: Mining big data: current status, and forecast to the future. ACM SIGKDD Explor. Newslett. **14**(2), 1–5 (2013)
10. Fausett, L.: Fundamentals of Neural Networks: an Architectures and Applications. Prentice Hall, New York (1994)
11. Hruschka, H., Natter, M.: Comparing performance of feed forward neural nets and k-means of cluster-based market segmentation. Eur. J. Oper. Res. **114**(3), 346–353 (1999)
12. Hung, C., Tsai, C.F.: Market segmentation based on hierarchical self-organizing map for markets of multimedia on demand. Expert Syst. Appl. **34**(1), 780–787 (2008)
13. Jang, S.C., Morrison, A.M.T., O'Leary, J.T.: Benefit segmentation of Japanese pleasure travelers to the USA and Canada: selecting target markets based on the profitability and the risk of individual market segment. Tour. Manag. **23**(4), 367–378 (2002)
14. Kahan, R.: Using database marketing techniques to enhance your one-to-one marketing initiatives. J. Consum. Mark. **15**(5), 491–493 (1998)
15. Kim, S.Y., Jung, T.S., Suh, E.H., Hwang, H.S.: Customer segmentation and strategy development based on customer lifetime value: a case study. Expert Syst. Appl. **31**(1), 101–107 (2006)
16. Kohonen, T.: Self-Organization and Associate Memory. Springer, Berlin (1984)
17. Kotler, P.: Marketing Management. Prentice-Hall, New York (2000)
18. Lee, J.H., Park, S.C.: Intelligent profitable customers segmentation system based on business intelligence tools. Expert Syst. Appl. **29**(1), 145–152 (2005)
19. Liang, Y.H.: Integration of data mining technologies to analyze customer value for the automotive maintenance industry. Expert Syst. Appl. **37**(12), 7489–7496 (2010)
20. McCarty, J.A., Hastak, M.: Segmentation approaches in data-mining: a comparison of RFM, CHAID, and logistic regression. J. Bus. Res. **60**(6), 656–662 (2007)
21. Pandys, A.S., Macy, R.B.: Pattern Recognition with Neural Networks in C++. CRC Press, Boca Raton (1996)
22. Pedrycz, W.: Granular Computing: Analysis and Design of Intelligent Systems, CRC Press, Boca Raton (2013)
23. Vellido, A.P., Lisboa, J.G., Meehan, K.: Segmentation of the on-line shopping market using neural networks. Expert Syst. Appl. **17**(4), 303–314 (1999)
24. Shin, H.W., Sohn, S.Y.: Product differentiation and market segmentation as alternative marketing strategies. Expert Syst. Appl. **27**(1), 27–33 (2004)
25. Smith, W.R.: Product differentiation and market segmentation as alternative marketing strategies. J. Mark. **12**, 3–8 (1956)
26. Tokunaga, H., Atlam, E.S., Fuketa, M., Morita, K., Tsuda, K., Aoe, J.I.: Estimating sentence types in computer related new product bulletins using a decision tree. Inf. Sci. **168**(1), 185–200 (2004)
27. Weiss, S.M.: Predictive Data Mining: A Practical Guide. Morgan Kaufmann, Burlington (1998)
28. Wu, M.L.: Application Practices of SPSS Statistics. Song-Gun Bookstore (2000)

Performance Competition for ISCIFCM and DPEI Models Under Uncontrolled Circumstances

Jui Fang Chang

Abstract The 2008 financial tsunami had a serious impact on the global financial industry. Thus, portfolio selection has become very important for individuals and companies to arrange their property and corporate financial management. High-return portfolios are usually accompanied by high risk; therefore, reducing risk and receiving remuneration are the main objectives. In short, constructing a satisfactory portfolio is very difficult. This paper proposes a new model to solve this problem. First, it uses the Investment Satisfied Capability Index and Fuzzy C-means Clustering (ISCIFCM) model developed by Chang and Chen (ICIC Express Lett 3(3):349–355, 2009) [4] and the DEA Portfolio Efficiency Index (DPEI) model proposed by Murthi et al. (Eur J Oper Res 98:408–418, 1997) [1] for stock selection in the securities market of Taiwan (Murthi et al. in Eur J Oper Res 98:408–418, 1997; Chang in Int J Organ Innov 2(3):225–249, 2010) [1, 3]. Then, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) are applied to these stocks to find the optimal investment allocation of the portfolio by using the moving interval windows. Comparing the performance of the models, the results show that the stock portfolio returns of the ISCIFCM model in our research is superior to the DPEI model. Its performance is also better than Taiwan Weighted Stock Index (TWSI) and Polaris Global ETFs (Exchange Trade Funds) Stable Fund in any period. The findings confirm that using better strategies for investors could provide improved performance.

Keywords ISCIFCM · DPEI · PSO · GA · Stocks portfolio

J. F. Chang (✉)
Department of International Business, National Kaohsiung University
of Applied Sciences, Kaohsiung 807, Taiwan, ROC
e-mail: rose@kuas.edu.tw

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_17,

1 Introduction

The liberalization of the global financial market over the recent years has led to a diversity of investments for investors. The large variety and quantity of financial products available in Taiwan, combined with rapid and complex flows of information, make it a daunting task for investors to select investment targets and achieve desired returns.

The stock market offers the most transparent and accessible information, compared to other investment vehicles. Meanwhile, it provides an opportunity to invest in different types of companies. The stock market is known for high liquidity and returns. It is hardly surprising that equity investments have always been popular with the general public and institutional investors. However, high returns are often accompanied by high risks. The global financial crisis in 2008 was a devastating blow to the financial industry around the world. Even the most experienced investment professionals can suffer heavy losses as a result of misreading the stock markets. Thus, achieving returns and minimizing risks is a critical issue to investors.

There is a wide selection of evaluation methods to gauge the performance of investment portfolios. Murthi et al. [1] constructed the DPEI on the basis of Data Envelopment Analysis (DEA) for the performance assessment of mutual funds. The concept is to evaluate investment efficiency of mutual funds by measuring trading costs. Moreover, the results aim to serve as a benchmark for investors to evaluate fund performances.

Regarding the overseas studies on funds of funds, Forthergill and Coke [2] sampled 450 funds of funds in Europe and examined their performances and risks. They suggested that funds of funds reduce systematic risks. If a given fund-of-funds portfolio contains 15–20 hedge funds, the risks will be comparable to fixed incomes and annual returns will stabilize in the 10–15 % range.

The application of artificial intelligence to investment has become a prevalent practice during the past years. Artificial intelligence is a tool to resolve the issues associated with asset allocations. Chang [3] used DPEI and genetic algorithms to construct a model for funds of funds. They suggested that the portfolio of investee funds chosen with DPEI offer better performances. Therefore, trading costs boast influence over returns [3]. The returns of these funds of funds are significantly better than TWSI as well as the funds of funds currently enjoying the best performance.

Chang and Chen [4] developed ISCIFCM, a new method for performance evaluations. ISCIFCM is the combination of Investment Satisfied Capability Index (ISCI) and Fuzzy C-means clustering Model (FCM) [4]. It not only defines the degree of satisfaction with investments but also empowers investors to quickly select outperforming stocks. The optimization technique in artificial intelligence, Particle Swarm Optimization (PSO), is used to assist investors to swiftly derive optimized solutions for investment portfolios and asset weights. The empirical results suggest that the stock portfolio constructed with ISCIFCM outperform TWSI and the portfolios chosen with fundamental analysis, the method most commonly used by investors nowadays.

This paper extends the efforts by Chang and Chen by exploring the application of ISCI, as well as by adding DEA Portfolio Efficiency Index for comparisons, in an effort to construct outperforming portfolios of stocks. It also applies GA [5–7] and PSO [8–11] for asset allocations, and seeks to identify which ones perform better with different investment portfolios. Moreover, stocks are selected using ISCIFCM. The goal is to determine whether the portfolio constructed in an unstable economy can outperform other funds. This paper also validates whether the equity portfolio built with ISCIFCM is superior to equity portfolios constructed with DPEI. A cross comparison is made with TWSI and Polaris Global ETFs Stable Fund. The purpose is to verify whether the equity portfolio constructed by this paper offers higher performance.

2 Research Method

This section firstly introduces DPEI model proposed by Murthi et al. [1], then describes the Investment Satisfied Capability Index and Fuzzy C-means Clustering (ISCIFCM) model developed by Chang and Chen [4].

2.1 DPEI

The inputs-oriented Charnes-Cooper-Rhodes (CCR) model is adopted for measurement. This model is based on DPEI for the comparison of performance by mutual funds proposed by Murthi et al. [1]. The DPEI model is described as follows:

$$\begin{aligned}\text{Max.}\quad DPEI &= \frac{R_0}{\sum_i \omega_i x_{i0} + \upsilon\sigma_0} \\ s.t.\frac{R_j}{\sum_{i=1}^{I} \omega_i x_{ij} + \upsilon\sigma_j} &\le 1, \qquad \omega_i, \upsilon \ge \varepsilon,\ \varepsilon > 0\end{aligned} \tag{1}$$

where i is the number of inputs, j is the number of stocks, R_0 is the annualized returns of the assessment period, R_j is the annualized return of the jth stock, x_{i0} is the P/E multiple, earnings growth, market capitalization, returns on equity and P/B multiple of individual stocks, x_{ij} is the ith ratio of the jth stock, σ_0 is the standard deviation of individual stocks, ω_i is the weight of the ith trading costs, υ is the weight of the standard deviation of returns, υ is a non-Archimedean constant, an infinite number smaller than any positive real number.

DPEI measures the returns accompanied with a certain level of risks and trading costs. It represents the returns of individual funds with market risks under control and trading expenses incurred. In other words, it quantifies the excess returns after the deduction of trading costs. How managers utilize resources to maximize output is a concept consistent with inputs/outputs in economic terms. Therefore, this concept is applied to the stock selection of investment portfolios.

2.2 ISCIFCM

This paper refers to ISCIFCM to evaluate the performance of stock portfolios. This index is suitable for the analysis of investments on one side. It has the advantage of a one-to-one mathematical relationship with the Investment Satisfied Index. Investors can quickly determine the performance of individual stocks by using this index and comparing the results with their desired returns.

Different investors have different levels of target returns (hence, different levels of satisfaction). They also set up the lower limit (LRL) for the required returns. As a general rule, investors demand a minimum return above the combination of risk-free interest rate and annualized increase of inflation. Therefore, this paper assumes *LRL* to be the sum of the risk-free rate and the annualized increase of inflation, as follows:

$$C_{SL} = \frac{\mu - LRL}{3\sigma} \tag{2}$$

where μ is the mean of the monthly return of stocks, σ is the standard deviation of the monthly return of stocks. The higher the upper bound of the LRL for indicator C_{SL}, the greater the monthly returns are. Meanwhile, the denominator of the index is the standard deviation of the monthly return of stocks. The smaller the σ value, the better the stability and the lower the variance of the stock is. The larger the C_{SL}, the better the stock performance. Obviously, C_{SL} is able to reasonably reflect the performance of individual stocks. Under the assumption of a normal distribution, the index is defined according to the level of satisfaction with investment returns, expressed as follows:

$$\begin{aligned} Pl &= P(X > LRL) = P(\frac{\mu - X}{3\sigma} < \frac{\mu - LRL}{3\sigma}) \\ &= P(-\frac{1}{3}Z < C_{SL}) = P(Z > \ -3C_{SL}) \\ &= 1 - \Phi(-3C_{SL}) = \Phi(3C_{SL}) \end{aligned} \tag{3}$$

Fuzzy C-means Clustering (FCM) is a clustering developed on the basis of C-means algorithm. It was first proposed by Bezdek [12] but is now being widely applied. The incorporation of fuzzy logic aims to improve the effectiveness of clustering [12].

As the name suggests, the biggest difference between FCM and K-means is the addition of the fuzzy concept. Data point *x* no longer belongs to a specific cluster. Rather, a value between 0 and 1 is used to represent the level of “belongingness” for x to a given cluster. It can be assumed that the expected number of clusters is $c\,(c_1, c_2, \ldots, c_c)$ and the whole data set includes n data points x $(x_1, x_2, x_3, \ldots, x_n)$. A matrix $U_{c\times n}$ is used to indicate the level of "belongingness" for each data point to each cluster. For any given point x_j in the data set, the sum of the levels of its "belongingness" to all the clusters will equal to 1.

$$\sum_{i=1}^{c} u_{ij} = 1, \quad \forall j = 1, 2, \ldots, n \tag{4}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{2}{m-1}}} \tag{5}$$

We define the target function J according to matrix U as follows:

$$J\left(U, c_{1,}c_2, \ldots, c_c\right) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c}\sum_{j=1}^{n} \left(u_{ij}\right)^m dist\left(c_i, x_j\right)^2 \tag{6}$$

where m is the weight coefficient, any value between 1 and infinity, $dist(c_i, x_j)$ is the distance function between c_i and x_j (usually the Euclidean distance). To meet the condition precedent in Eq. (4), the new target function J_{new} is defined according to Eq. (6) as follows:

$$\begin{aligned} &J_{new}(U, c_1, c_2, \ldots, c_c, \lambda_1, \lambda_2, \ldots, \lambda_n) \\ &\quad = J(U, c_1, c_2, \ldots, c_c) + \sum_{j=1}^{n} \lambda_j \left(\sum_{i=1}^{c} u_{ij} - 1\right) \\ &\quad = \sum_{i=1}^{c}\sum_{j=1}^{n} \left(u_{ij}\right)^m dist\left(c_1, x_j\right)^2 + \sum_{j=1}^{n} \lambda_j \left(\sum_{i=1}^{c} u_{ij} - 1\right) \end{aligned} \tag{7}$$

where λ_j is the Lagrange multiplier in the restricted set in Eq. (7). To achieve the optimization solution of J_{new}, we differentiate the parameters and propose the following conclusions:

$$c_i = \frac{\sum_{j=1}^{n} \left(u_{ij}\right)^m X_j}{\sum_{j=1}^{n} \left(u_{ij}\right)^m} \tag{8}$$

3 Research Design

The design of this study includes two experiments. Experiment 1 utilizes ISCIFCM to select stocks, whereas Experiment 2 uses DPEI for stock picks from blue chips listed on Taiwan Stock Exchange (TSE) and Gre Tai Securities Market. These experiments are in two stages: stock selections for the first stage and capital allocations for the second stage. The allocations of the stocks chosen during the first stage are based on GA and PSO, in order to construct optimized portfolios.

3.1 Experiment 1: Stock Selection with ISCIFCM

3.1.1 Stock Selection

The procedures of choosing stocks with ISCIFCM during the first stage are as follows:

Step 1: Screening of blue chips listed in Taiwan with C_{SL}

$$C_{SL} = \frac{\mu - LRL}{3\sigma} \tag{9}$$

where μ is the mean of the monthly return of stocks, and σ is the standard deviation of the monthly return of stocks. LRL is the 210-day treasury interest rates plus the annualized increase of inflation.

Step 2: Feed the calculated annual inputs by using C_{SL} into Matlab clustering program and pick 15 stocks by referring to literature.

3.1.2 Capital Allocation

In the second stage of the experiment, GA and PSO are used for capital allocations (weight allocations) of the 15 stocks chosen with ISCIFCM for risk diversification and portfolio construction.

3.2 Experiment 2: Stock Selection with DPEI

3.2.1 Stock Selection

In the first stage of the experiment, the stocks are chosen with DPEI. An inputs-oriented CCR model is constructed on the basis of DPEI developed by Murthi et al. [1] for the evaluation of mutual fund performances. Stocks are selected by using the five factors most frequently used by fund managers. The inputs are the expense ratio of different years, and the output is the mean return of individual stocks across the sampling years.

3.2.2 Capital Allocation

In the second stage of the experiment, GA and PSO are used for capital allocations (weight allocations) of the 15 stocks chosen with DPEI for risk diversification and portfolio construction.

3.2.3 Parameter Set-Up

1. GA parameter set-up
   (a) Initialization: 160.
   (b) Coding: Real number coding is adopted because it is consistent with the data type. This saves processing time, enhances the accuracy of the system and boosts the possibility of arriving at the optimal solution.
   (c) Selection: roulette wheel selection at a selection probability of 0.5.
   (d) Crossover: 0.5; two-point crossover.
   (e) Mutation: 0.03.
   (f) Termination condition: 5,000 generations.
2. PSO parameter set-up
   (a) Cluster: 160.
   (b) Maximum speed: $v_{\max} = 4$ based on literature; this means the maximum range of speed for each generation is 8.
   (c) Learning factor: $c_1 = c_2 = 2.0$ based on literature.
   (d) Inertia weight: initial weight at 0.9, on a linear sliding scale; final weight at 0.4.
   (e) Termination: terminated at 5,000 generations.

3.2.4 Fitness Function

$$\begin{aligned} &\text{Max } F = \left\{ \sum_{i=1}^{n} \left[v_i \bar{r}_i - \left[\sqrt{\left(\sum_{j=1}^{n} v_j^2 S_j^2 + \sum_{j=1}^{n} 2 v_i v_j S_{ij} \,|_{i \neq j} \right)} \right] \right] \right\} \\ &s.t. \quad 0 < v_i \leq 1.0, \sum_{i=1}^{n} v_i = 1 \end{aligned} \tag{10}$$

where F is the fitness function, v_i is the allocated weight of the ith asset, S_j^2 is the variance of the jth stock in the portfolio, $\bar{r}_i$ is the historical mean return of the ith stock, n is the number of assets in the portfolio, S_{ij} is the co-variance between the ith asset and the jth asset.

3.2.5 Data Collection and Sampling Period

This paper samples companies listed on Taiwan Security Exchange (TSE) and GreTai Securities Market. Companies with incomplete data are eliminated from the sampling pool. Data is sourced from the returns of stocks and standard deviations of historical returns in the database of Taiwan Economic Journal, as well as from basic and financial information of individual stocks on the website of MasterLink Securities Corp. The sampling period is from January 1, 2006 to December 31, 2009. Different stocks can be selected for different periods given the research purpose of achieving greater returns across periods by using GA and PSO.

4 Experiment Analysis

This paper constructs investment portfolios ISCIFCM_GA, ISCIFCM_PSO, DPEI_GA and DPEI_PSO, and conducts a cross comparison on the returns in a Stable Economy Period and an Unstable Economic Period. In addition to the comparison of the returns of the portfolios built with different models, the results with TWSI and Polaris Global ETF are compared to examine whether stock-selection models are subject to the influence of the TAIEI levels.

4.1 Stable Economy Period: ISCIFCM and DPEI_2006 Stock Selections for Forecasts in 2007

The stocks are selected from 2006, with a training period built with three-month data. The testing period is built with one-month data for the rolling window analysis. GA and PSO are applied to the capital allocations for the ISCIFCM portfolio and the DPEI portfolio. Then, four portfolios are built, namely ISCIFCM_GA, ISCIFCM_PSO, DPEI_GA, and DPEI_PSO. The results are compared with TWSI and Polaris Global ETF of the same years as benchmarks.

According to Table 1, the annualized returns of the ISCIFCM portfolio are higher than the DPEI portfolio, TWSI and Polaris Global ETF. The ISCIFCM_PSO portfolio, with returns of 33.18 %, outperforms ISCIFCM_GA, with returns of 23.43 %. The DPEI_PSO portfolio yields a negative return of 9.58, beating DPEI_GA, with a negative return of −13.92 %. Figure 1 shows the ISCIFCM portfolios outperforming the DPEI portfolios, TWSI and Polaris Global ETF in most months (with the returns in May and December slightly lower than those of other portfolios). In fact, the ISCIFCM portfolios significantly outperform others in January, March, July and August. The results suggest that PSO is a better choice than GA for capital allocations based on the annualized forecasted returns in 2007 of the stocks selected in 2006.

Table 1 2006 stock selections—comparison of ISCIFCM, DPEI, TWSI and polaris global ETF

Portfolio Date	ISCIFCM GA	ISCIFCM PSO	DPEI GA	DPEI PSO	TWSI	Polaris global ETF
2007.1	3.68	5.96	−12.51	−12.47	−1.59	0.09
2007.2	3.50	5.39	3.59	2.49	2.63	2.62
2007.3	12.05	11.83	−3.52	−4.98	−0.22	−0.94
2007.4	−0.77	0.71	6.43	5.88	−0.11	2.75
2007.5	−9.50	−8.82	10.44	10.95	3.42	−0.84
2007.6	4.86	2.74	3.88	4.00	9.06	−0.76
2007.7	12.52	18.06	8.80	10.66	4.55	−0.26
2007.8	14.98	13.40	0.19	2.07	−3.29	−0.68
2007.9	4.40	6.81	1.59	1.65	5.50	4.73
2007.10	−6.36	−8.69	−11.66	−11.18	2.48	1.80
2007.11	−14.33	−11.02	−20.30	−19.67	−11.58	−1.93
2007.12	−1.61	−3.19	−0.85	1.02	−0.93	0.66
Annualized returns	23.43	33.18	−13.92	−9.58	9.92	7.24

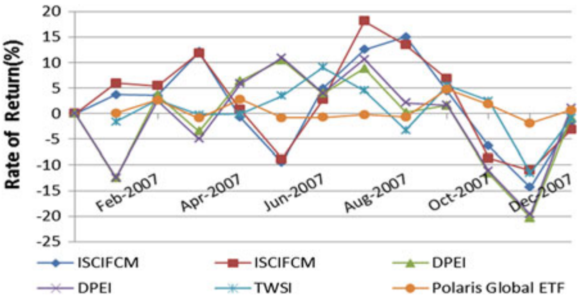


Fig. 1 2006 stock selections—comparison of ISCIFCM, DPEI, TWSI and polaris global ETF

4.2 *Unstable Economic Period: ISCIFCM and DPEI_2008 Stock Selections for Forecasts in 2009*

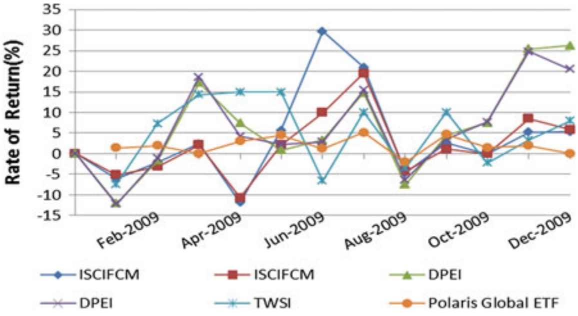
The stocks are selected from 2008, with a training period based on three-month data. The testing period is built with one-month data for the rolling window analysis. GA and PSO are applied to the capital allocations for the ISCIFCM portfolio and the DPEI portfolio. Hence, four portfolios are built, namely ISCIFCM_GA, ISCIFCM_PSO, DPEI_GA, and DPEI_PSO. The results are then compared with TWSI and Polaris Global ETF of the same years as benchmarks.

According to Table 2 (stock selections in 2008), the annualized returns of the ISCIFCM portfolio are higher than Polaris Global ETF, but lower than the DPEI portfolio and TWSI. As far as GA and PSO as the capital allocation methods are concerned, ISCIFCM_GA (returns at 46.95 %) outperform ISCIFCM_PSO (returns at 25.66 %); likewise, DPEI_GA (returns at 85.76 %) outperforms DPEI_PSO (returns at 79.93 %). Figure 2 (stock selections in 2008) compares the

Table 2 2008 stock selections—comparison of ISCIFCM, DPEI, TWSI and polaris global ETF

Portfolio Date	ISCIFCM GA	ISCIFCM PSO	DPEI GA	DPEI PSO	TWSI	Polaris global ETF
2009.1	−6.10	−5.18	−12.11	−12.30	−7.48	1.43
2009.2	−2.19	−3.18	−1.74	−1.13	7.28	1.94
2009.3	2.40	2.16	17.27	18.57	14.34	0
2009.4	−11.93	−10.74	7.37	4.19	15.00	3.03
2009.5	5.65	2.13	0.87	2.18	14.98	4.45
2009.6	29.72	9.91	3.35	2.88	−6.65	1.21
2009.7	20.98	19.51	14.76	15.40	10.04	5.16
2009.8	−4.69	−4.26	−7.57	−6.34	−3.56	−2.04
2009.9	2.62	1.08	4.43	3.56	10.01	4.70
2009.10	−0.04	−0.08	7.48	7.65	−2.25	1.47
2009.11	5.22	8.47	25.40	24.77	3.30	1.96
2009.12	5.29	5.82	26.25	20.51	7.99	0
Annualized returns	46.95	25.66	85.76	79.93	63.00	23.31

Fig. 2 2008 stock selections—comparison of ISCIFCM, DPEI, TWSI and polaris global ETF



returns of ISCIFCM, DPEI, TWSI and Polaris Global ETF. The ISCIFCM portfolio constructed by this paper outperforms Polaris Global ETF, but underperforms the DPEI portfolio and TWSI. Despite the relative underperformance, the annualized returns on the stock selections in 2008 for 2009 forecasts show that DPEI_GA reports the best capital allocations. This may be a result of the global financial crisis in 2008 (Unstable Economic Period). When stocks are chosen in 2008 for the returns in 2009, the ISCIFCM model reflects the poor economy in 2008 with investors unwilling to allocate significant amount of capital to the stock market. As a result, the portfolio outperforms Polaris Global ETF but underperforms DPEI and TWSI.

4.3 Effects of Economic Changes on Performance of Stock Picks in Different Periods

This section explores whether the effects of the global financial crisis in 2008 show any significant variances on investment returns. The performances of the ISCIFCM model and the DPEI models are compared before and after 2008 under different capital allocations.

4.3.1 Stable Economic Period

This section compares the investment portfolios for returns in 2007 defined as Stable Economic Period.

According to Table 3, ISCIFCM outperforms DPEI with capital allocations based on GA and PSO. It also yields higher forecasted returns than TWSI and Polaris Global ETF during the same year. Generally speaking, the stock selections based on ISCIFCM outperform those based on DPEI in a stable economy, regardless of whether capital allocations use GA or PSO.

4.3.2 Unstable Economic Period

This section compares the investment portfolios for returns in 2009 defined as Unstable Economic Period.

Table 4 shows that the expected returns of the DPEI portfolio are higher than the ICIFCM portfolio in the Unstable Economic Period, TSWI and Polaris Global ETF.

The performances of the portfolios are compared with different capital allocations and reveal that in the Unstable Economic Period, the ISCIFCM_GA model outperforms the ISCIFCM_PSO. This indicates stronger predicative power of GA.

Based on the above conclusions, this paper infers that the ISCIFCM stock selections are influenced by the subjective judgment of investors in the Unstable Economic Period. When share price movements are volatile, investors are unwilling to invest a large amount in the stock market. Also, the ISCIFCM model is subject to the influence of the levels of satisfaction with investments. A volatile stock market means that the minimum returns demanded by investors are higher than risk-free rates and the annualized increase of inflation. Meanwhile, the ISCIFCM portfolio constructed by this paper outperforms Polaris Global EFP (as the minimum expected return). This proves that investors withdraw from the markets once they have achieved the desired returns (beating the minimum expected returns) and will not take further risks by keeping money in the stock market.

The outperformance of the DPEI portfolios in Unstable Economic Period is perhaps because the inputs/outputs chosen by this paper are financial ratios. Under normal circumstances, financial ratios are stable, and do not change significantly even in a recession. Therefore, better companies can weather out the bad times (i.e. a volatile stock market) and endure to the recovery (rebound of the stock market).

Table 3 Comparison of investment portfolios for returns in stable economic period of 2007

ISCIFCM_forecasted for 2007	ISCIFCM returns	DPEI returns	DPEI_forecasted for 2007
ISCIFCM_GA expected returns (2006 stock selections)	23.43	−13.91	DPEI_GA expected returns (2006 stock selections)
ISCIFCM_PSO expected returns (2006 stock selections)	33.18	−9.58	DPEI_PSO expected returns (2006 stock selections)
2007 TWSI	9.92	9.92	2007 TWSI
2007 Polaris Global ETFs Stable Fund	7.24	7.24	2007 Polaris Global ETFs Stable Fund

Table 4 Comparison of investment portfolios for returns in instable economy period of 2009

ISCIFCM_forecasted for 2009	ISCIFCM returns	DPEI returns	DPEI_forecasted for 2009
ISCIFCM_GA expected returns (2008 stock selections)	46.95	85.78	DPEI_GA expected returns (2008 stock selections)
ISCIFCM_PSO expected returns (2008 stock selections)	25.66	79.93	DPEI_PSO expected returns (2008 stock selections)
2009 TWSI	63	63	2009 TWSI
2009 Polaris Global ETFs Stable Fund	23.31	23.31	2009 Polaris Global ETFs Stable Fund

5 Conclusion

This paper samples the four-year historical data from 2006 to 2009 for the training and testing of stock picks. The global financial crisis in 2008 is used as a watershed to divide the sampling periods into two sub-periods, namely Stable Economic Period (with stock selections in 2006 for expected returns in 2007) and Unstable Economic Period (with stock selections in 2008 for expected returns in 2009).

The empirical results suggest that ISCIFCM is superior to DPEI as a stock selection model in a stable economy, and performs better than TWSI and Polaris Global ETF. However, in an unstable economy, the expected returns of the DPEI portfolio are higher than the ISCIFCM portfolio, TWSI and Polaris Global ETF. As far as different capital allocations are concerned, ISCIFCM_PSO outperforms ISCIFCM_GA in a stable economy. Moreover, DPEI_GA outperforms DPEI_PSO in an unstable economy. The possible reason may be the financial ratios used as inputs/outputs. Under normal circumstances, financial ratios are quite stable, and do not change significantly even in a recession. Therefore, stronger companies can survive the bad times (i.e. a volatile stock market) and rebound from the state of depression.

References

1. Murthi, B.P.S., Choi, Y.K., Desai, P.: Efficiency of mutual funds and portfolio performance measurement: a non-parametric approach. Eur. J. Oper. Res. **98**, 408–418 (1997)
2. Fothergill, M., Coke, C.: Funds of hedge funds: an introduction to multi-manager funds. J. Altern Investments **4**(2), 7–16 (2001)
3. Chang, J.F.: DEA investment portfolio efficiency index based on GA applied in the establishment of a fund of funds. Int. J. Organ. Innov. **2**(3), 225–249 (2010)
4. Chang, J.F., Chen, K.L.: Applying new investment satisfied capability index and particle swarm optimization to construct stock portfolio. ICIC Express Lett. **3**(3), 349–355 (2009)
5. Holland, J.H.: Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor (1975)
6. Holland, J.H.: Genetic algorithms. Sci. Am. **267**, 66–72 (1972)
7. Chang, J.F.: Performance comparison between genetic algorithms and particle swarm optimization in constructing equity portfolios. Int. J. Innov. Comput. Info. Control **5**(12), 5069–5079 (2009)
8. Eberhart, R.C., Shi, Y.: Particle swarm optimization: developments, applications and resources. In: Congress Evolutionary Computation 2001. IEEE service center, Piscataway, NJ, Seoul, Korea (2001)
9. Lin, Y., Qin, Z., Shi, Z., Lu, J.: Center particle swarm optimization. Neurocomputing **70**, 672–679 (2006)
10. Yoshida, H., Kawata, K., Fukuyama, Y., Nakanishi, Y.: A particle swarm optimization for reactive power and voltage control considering voltage stability. In: Proceedings of International Conference on Intelligent System Application to Power Systems, pp. 117–121. Rio de Janeiro, Brazil, (1999)
11. Chang, J.F., Shi, P.: Using investment satisfaction capability index based particle swarm optimization to construct a stock portfolio. Inf. Sci. **181**(14), 2989–2999 (2011)
12. Bezdek, J.: Cluster validity with fuzzy sets. J. Cybern. **3**(3), 58–78 (1973)

Rough Set Model Based Knowledge Acquisition of Market Movements from Economic Data

Yoshiyuki Matsumoto and Junzo Watada

Abstract The concept and method of rough sets were proposed by Z. Pawlak in 1982. This method enables us to mine knowledge granules as decision rules from a database, a web base, a set and so on. The obtained decision rules can be applicable for data analysis as well as used to reason, estimate, evaluate, or forecast an unknown object. The objective of this paper is to apply the rough set method to time series data for mining knowledge granules, and especially to mine knowledge granules from the data set of tick-wise price fluctuations.

Keywords Rough set model · Knowledge acquisition · Market movement · Economic data · Knowledge granule · Decision rule · Data analysis · Tick-wise price

1 Introduction

The analysis of time-series data is widely used in corporations and economics. Especially, the technical analysis is used to model the change of prices based on the graphical expression of market movement and the fundamental analysis is applied to understand the corporate performance and economic environment of a company. Also Y. Matsumoto and J. Watada employ the chaotic method to forecast the future price value [1]. This paper employs rough sets method to analyze time-series data [2, 3]. The rough sets analysis is proposed by Z. Pawlak to acquire rule-based

Y. Matsumoto
Shimonoseki City University, 2-1-1, Daigaku-Cho, Shimonoseki
Yamaguchi 751-8510, Japan
e-mail: matsumoto@shimonoseki-cu.ac.jp

J. Watada (✉)
Waseda University, 2-7 Hibikino, Wakamatsu-ku, Kitakyushu
Fukuoka 808-0135, Japan
e-mail: junzow@osb.att.ne.jp

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_18,

knowledge from a set defined with plural attributes. The objective of this paper is to mine the knowledge granules of price movement from the intra-day trading data of stock prices called tick. Each dealt price is recorded whenever the dealing is accomplished. This paper aims to mine the knowledge of forecasting from the tick data, and using the knowledge gained, we predicted stock prices.

2 Rough Sets Theory

A rough set is especially useful for domains where the data collected are imprecise and/or incomplete about the domain objects. It provides a powerful tool for a data analysis and data mining of imprecise and ambiguous data. A reduction is the minimal set of attributes that preserves the indispensability relation, that is, the classification power of the original dataset [4]. Rough set theory has many advantages, such as providing efficient algorithms for finding hidden patterns in data, finding minimal sets of data (data reduction), evaluating the significance of data, and generating the minimal sets of decision rules from data. It is easy to understand and to offer a straightforward interpretation of the results [5]. These advantages can simplify analyses, which is why many applications use a rough set approach as their research method. The rough set theory is of fundamental importance in artificial intelligence and cognitive science, especially in the areas of machine learning, knowledge acquisition, decision analysis, knowledge discovery from databases, expert systems, decision support systems, inductive reasoning, and pattern recognition [6, 7, 8].

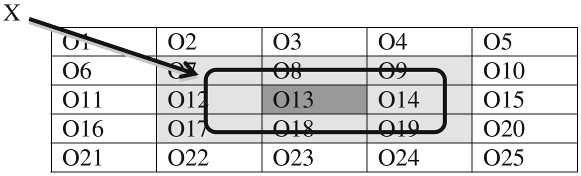
Rough set theory has been applied to the management of many various issues, including expert systems, empirical study of materials data [9], machine diagnosis [10], travel demand analysis [11], web screen design [12], IRIS data classfication [13], business failure prediction, solving linear programs, data mining [14] and α-RST [15]. Another paper discusses the preference-order of the attribute criteria needed to extend the original rough set theory, such as sorting, choice and ranking problems [16], the insurance market [17], and unifying rough set theory with fuzzy theory [18]. Rough set theory provides a simple way to analyze data and reduct information.

2.1 Information Systems

Generally, an information system denoted IS is defined as $IS = (U, A)$, where universe U consists of finite objects and A is named a universe and A is a finite set of n attributes $\{a_1, a_2, \ldots, a_n\}$. Each attribute a belongs to set A, that is, $a \in A$. An object ω (ω ϵ U) has a value $f_a(\omega)$ for each attribute, which is defined as f_a: $U \rightarrow V_a$. f_a means that object ω in the U has a value $f_a(\omega) \in V_a$ for attribute $a \in A$, where V_a is a set of values of attribute $a \in A$. It is called a domain of attribute a (Table 1).

Table 1 Sample Information System

Object	P1	P2	P3	P4
O1	+	−	+	−
O2	−	−	−	+
O3	+	+	+	+
O4	−	−	+	−
O5	−	+	−	−



O1	O2	O3	O4	O5
O6	O7	O8	O9	O10
O11	O12	O13	O14	O15
O16	O17	O18	O19	O20
O21	O22	O23	O24	O25

Fig. 1 Upper and lower approximations

2.2 Lower and Upper Approximations

Rough sets analysis is based on the two basic concepts: lower and upper approximations of a focal set. The upper approximations is the set of all elements which possibly belong to the focal set and the lower approximations of all elements which doubtlessly belong to the focal set. Let X be a subset of elements in universe U, that is, $\mathrm{X} \subset \mathrm{U}$. Let us consider a subset in V_a, $P \subseteq V$.

The low approximation of P, denoted PX, is the union of all equivalence classes which are contained by the target set as follows: $\underline{PX} = \{x|[x]_P \in X\}$.

The upper approximation of P, denotes $\overline{PX}$, is the union of all equivalence classes which have non-empty intersection with the target set as follows: $\overline{PX} = \{x|[x]_P \cap X \neq \phi\}$.

The boundary set of X in U is defined as following: $PNX = \overline{PX} - \underline{PX}$ (Fig. 1).

2.3 Decision Ruless

An information system denoted IS is defined as $IS = (U, A)$, A can be partitioned into two disjoint classes $C, D \subseteq A$ of attributes, called condition and decision attributes, respectively. Here $IS = (U, C, D)$ is called decision system.

Decision rules can also be regarded as a set of decision (classification) rules of the form: $a_k \rightarrow d_j$, where a_k means that attribute a_k has value 1, d_j means the decision attributes and the symbol → denotes propositional implication. In the decision rule $\theta \rightarrow \varphi$, formulae θ and φ are called condition (premise) and decision (conclusion), respectively [19]. The decision rules we can minimize the set of attributes, reduce the superfluous attributes and classify elements into different groups. In this way we can have many decision rules, each rule shows meaningful attributes. The stronger rule will cover more objects and the strength of each decision rule indicate the appropriateness of rules.

2.4 Analysis of Decision Rules

Only decision rules that are obtained rough set theory and have high C.I. are employed in reasoning. C.I. is an abbreviation of Covering Index that is a rate of objects that can sufficiently reach the same decision attribute by the rule out of the whole objects [20]. If whole objects number is 5, corresponding objects number is 3, C.I. is 0.6.

Generally speaking, decision rules with high C.I. are highly reliable and results in good reasoning. In real situations, the number of obtained decision rules is often more than several hundreds. In these cases, reasoning can not employ almost all decision rules. That is, reasoning scattered almost decision rules.

It is necessary to make decision rules effective so as to combine decision rules by means of decision rule analysis [21]. Decision rule analysis enables us to obtain new combined decision rules by means that premises of decision rules are decomposed and given some points depending on their C.I. value. This method enables us to take all decision rules into consideration even if rules have a low C.I. value. In this paper, decision rules are combined and applied to forecasting

Let us explain the detail of decision rule analysis. The decision rule analysis determines rules by calculating their column scores. The column score can be calculated in the following:

Let us consider the following three rules.

$$\begin{gathered}\text{IF a} = 1 \text{ and } \text{b} = 1 \text{ then d} = 1(\text{C.I.} = 0.4)\\ \text{IF b} = 2 \text{ then d} = 1(\text{C.I.} = 0.3)\\ \text{IF a} = 2 \text{ and b} = 2 \text{ and c} = 1 \text{ then d} = 1(\text{C.I.} = 0.6)\end{gathered} \tag{1}$$

The column score can be obtained using the combination table as shown in 0. The combination table is an n × n matrix consisting of all attributes. An element of the combination table is a score of combination of two attributes.

For example, the first rule has a = 1 and b = 1 as its premises. On this case, the vertical column has a = 1 and the horizontal row has b = 1, and the vertical column has b = 1 and horizontal row has a = 1. We describe two scores in these elements. The score value is one or C.I. value divided by the written score value.

On this case, two elements have each score value.

$$0.4/2 = 0.2 \tag{2}$$

On the case of the second rule, as the premise has one attribute, the column and row are written 0.3 for b = 2.

Table 2 Combination table

	a = 1	a = 2	b = 1	b = 2	c = 1	c = 2	Column score
a = 1			0.2				0.2
a = 2				0.1	0.1		0.2
b = 1	0.2						0.2
b = 2		0.1		0.3	0.1		0.5
c = 1		0.1		0.1			0.2
c = 2							

On the case of the third rule, as the premise has 3 attributes, 6 elements ($_3C_2 = 6$) should be written scores. The written score is

$$0.6/6 = 0.1 \tag{3}$$

The column score is the total value of scores in each column. For example, on the case of a = 2 we obtain

$$0.1 + 0.1 = 0.2 \tag{4}$$

This calculation results in Table 2. Using this combination we can derive a decision table. For example, on the case of column b = 2, since there is a score in a = 2, b = 2 and c = 1, the rule of this column results in as follows:

$$\text{IF a} = 2 \text{ and b} = 2 \text{ and c} = 1 \text{ then d} = 1 \tag{5}$$

Usually, scores under the some threshold are not accepted. For instance, when the threshold is 0.2, the rule is written in the following:

$$\text{IF b} = 2 \text{ then d} = 1 \tag{6}$$

3 Regression Line-Base Analysis

In general, rough sets analysis deals with categorical data. Therefore, in this paper we obtain regression line for the time-series data to forecast future values and use the up and down trends of the regression line as a condition attribute. For example, when we analyze past six fiscal terms, we can obtain the trends of the regression line for all six terms, the former three terms and the latter three terms. The trend *a* is decided as follows:

$$a = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^{n} (X_i - \bar{X})^2}$$

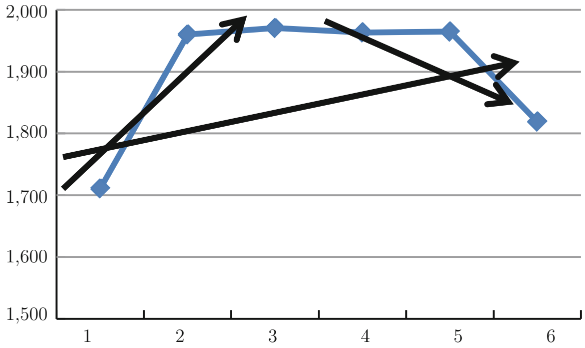


Fig. 2 Trends by linear regression

Table 3 Trends of data

No.	Condition attribute (Trend)			Decision attribute
	Total term	Former term	Latter term	Present term
1	+	+	−	−
2	+	−	+	+
3	+	−	−	−
4	+	−	−	+
5	−	−	+	−

The obtained trend *a* is employed as a condition attribute. That is, relating to each of the total, former and latter parts we forecasted whether each part has increasing or decreasing trend, depending on such data. On the case as shown in Fig. 2, the condition attribute shows that the total trend shows plus trend, the former part has plus trend and the latter part indicates minus trend. Such values are evaluated concerning with each of data, then we mine the knowledge by using rough sets analysis. Table 3 illustrates this process.

According to this process, we analyzed the trend of each of past data and forecasted the up and down movements of the present term depending on the data.

4 Rough Sets-Based Knowledge Acquisition of Market

In this research study, we analyzed the time series data of the market by using the regression line, and knowledge acquisition from the result by using rough sets. Using the regression line, the trend of the data can be understood, And it is possible to predict the state after a certain time. The analysis used the company's stock price data of the Tokyo stock market. We acquired knowledge granules by using the stock price of Fuji Heavy Industries on June 02, 2008. The original data are tick data, in this analysis are used to acquire the knowledge granules to create a 1 min chart data from its data. We were using data that has been trading in between 9 and 11 a.m. Knowledge acquisition is done for the trend data of up to previous 12 min. We are classified the data into seven types (whole, first half, second half, first quarter, second quarter, third quarter, fourth quarter). We analyzed the trend of rising and falling for seven types. The decision attribute is whether the value increased or decreased after 1 min. We acquired the knowledge granule to predict the increase or decrease movement of 1 min using the previous 12 min data. We are creating a decision table. Objects in the decision table is a trend in each time. Conditions attributes of the object is the trend of the past. Decision attribute of the object is the trend of the future.

4.1 Result of Knowledge Acquisition

Tables 4, 5 and 6 show decision rules acquired by using a rough set. C.I. is an abbreviation of the Covering Index showing the proportion of the target corresponding to that rule in the target with the same decision attribute [15]. The rule is reliable means that the C.I. value is higher. We have presented only the top 10 rules of the C.I. "+" in this table means that this period was increasing. "−" shows the descent, "0" indicates no change as well.

Table 4 shows the rule was a descent in this period. If the whole was descent, it shows that the descent could be high after 1 min. Even if the second half rise and fourth quarter was descent or no change, it shows that the value of the current period was descent.

Table 5 shows the rule was no change in this period. This table includes a lot of "0". If the stock price does not change much in 12 min, which indicates no change after 1 min. Also, "−" did not exist at all in knowledge granules of the data. If the increasing trend was found before 12 min, this period had often no change.

Table 6 shows the rule was an increase in this period. This table shows that the often descent to first and third period, also increase to second and fourth period (Fig. 3).

4.2 The Prediction Based on the Knowledge Acquired

In this section, we predicted the stock price on the basis of the knowledge acquired through the rough set. The stock to very short term fluctuations, we assume affected the behavior of the past. We used the rules obtained from the transaction data in the morning of that day, and predicted the price movements based on previous 12 min after the start of the afternoon trading. The regression line is obtained using the same data that were predicted. We predicted the price movements from the start of trading before the slope of the regression line became "rising" or "falling".

Table 4 Rules of decreasing cases

No.	Decision rule							C.I.
	Whole	Former	Latter	First quarter	Second quarter	Third quarter	Fourth quarter	
1	−	−		−			0	0.143
2	−		+				0	0.095
3	−		+				−	0.095
4		+		−			−	0.095
5		−		0		0		0.095
6	−		+	+				0.095
7			+	+	−			0.095
8	+			−			−	0.095
9		−		−		−	0	0.095
10	−	−			0		0	0.095

Table 5 Rules of no changing cases

No.	Decision rule							C.I.
	Whole	Former	Latter	First quarter	Second quarter	Third quarter	Fourth quarter	
1	+			0			0	0.217
2				0		+	0	0.117
3		+	+		+			0.117
4		+		0		0		0.117
5			+		+		0	0.100
6	+		0		0			0.100
7		+			0	0	0	0.100
8	+				0	0	0	0.100
9		+	0		0	0		0.100
10		+	0		0		0	0.100

Table 6 Rules of increasing cases

No.	Decision rule							C.I.
	Whole	Former	Latter	First quarter	Second quarter	Third quarter	Fourth quarter	
1		+		0		−		0.211
2		+		0			−	0.158
3			+	−	0			0.158
4			−	0	+			0.158
5				0	+	−		0.158
6				−		−	+	0.105
7				−	0		+	0.105
8		−			0		+	0.105
9	+			0		−		0.105
10	−			0	+			0.105

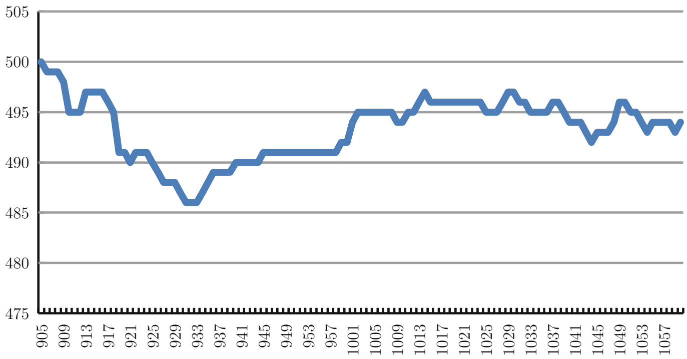


Fig. 3 One-minute chart of Fuji heavy industry

Table 7 The number of hitted rules in the afternoon

Time	Number of rules			Prediction
	Down	No change	Up	
12:30	0	3	3	Up
12:31	2	3	4	Up
12:32	1	5	3	No change
12:33	0	2	0	No change
12:34	0	8	2	No change
12:35	1	5	4	No change
12:36	0	2	6	Up
12:37	0	5	0	No change
12:38	1	4	2	No change
12:39	0	0	0	No change
12:40	4	0	0	Down
12:41	0	0	0	No change

Table 7 shows the number of rules that were used to predict the price movements of 12 min from the start of the afternoon trading. For example, this row of 12:30 shows that the number of hits to descending rule is zero, no change rule is three, increasing rule is three. We used as the predicted value the state often hits the most.

4.3 Prediction Result

We have predicted the stock price of Fuji Heavy Industries Ltd. by using the proposed method. The data used in the prediction are stock prices that were traded on June 2008. We have predicted the stock price by using a decision table in rough set. Objects in the decision table is a trend at each time. We have calculated the minimum decision rules from the decision table. Minimal decision rule is a knowledge of the prediction in specific conditions.

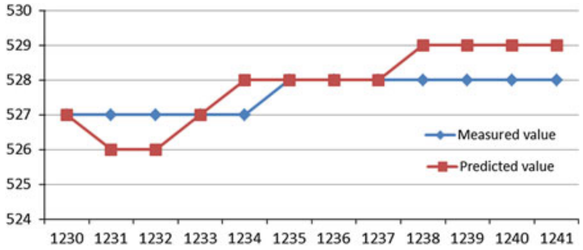


Fig. 4 The change of real and predicted values (1)

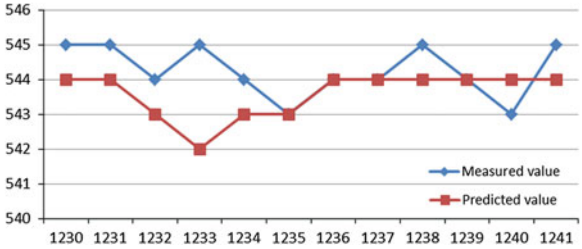


Fig. 5 The change of real and predicted values (2)

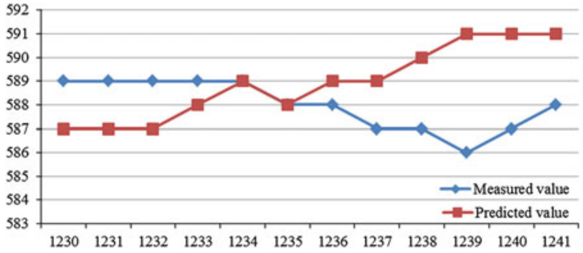


Fig. 6 The change of real and predicted values (3)

The knowledge of predictions obtained, it is to predict the stock price.

Figures 4, 5 and 6 show a part of the predicted results.

The measured values in Fig. 4 show a moderate upward trend. In addition, predicted values are all in the same general price movements. This figure shows that the prediction was correct.

Figure 5 shows that both values were vibrating and did not change much.

The measured values in Fig. 6 show a moderate downward trend. However, the predicted value was an upward trend. This figure shows that the prediction was not correct.

Table 8 is a summary of the prediction results of one month. The slope of the regression line is described as " + " for positive, described as " − " for negative. It is predicted using the knowledge granules obtained from trading in the morning. We predicted the 12 min immediately after the start of trading in the afternoon. Its value was predicted whether to fall or rise. The correct rate of predicted values in one month was 61.9 %.

Table 8 The rate of correct predictions in the month

Date	Measured value	Predicted value	Correct predictions
20080602	+	−	×
20080603	+	+	○
20080604	−	+	×
20080605	+	+	○
20080606	+	−	×
20080609	+	+	○
20080610	−	−	○
20080611	+	−	×
20080612	+	−	×
20080613	+	+	○
20080616	+	+	○
20080617	+	+	○
20080618	+	+	○
20080619	−	+	×
20080620	+	+	○
20080623	+	+	○
20080624	+	+	○
20080625	+	−	×
20080626	+	+	○
20080627	+	+	○
20080630	−	+	×
			61.9 %

Table 9 Toyota and Fuji heavy industry

Date	Comparison of the measured value	Date	Comparison of the measured value
20080602	○	20080617	○
20080603	○	20080618	○
20080604	○	20080619	×
20080605	×	20080620	×
20080606	×	20080623	○
20080609	○	20080624	○
20080610	○	20080625	○
20080611	○	20080626	○
20080612	○	20080627	○
20080613	○	20080630	○
20080616	○		81.0 %

Stock prices are affected by the stock price of other related stocks. Therefore, we propose to acquire knowledge granules from stock price data of other related stocks. Prediction was performed by using the movement of the relevant stocks. We used Daihatsu and Toyota as related stock prices. We acquired the knowledge of the prediction for stock prices of these two companies.

Table 9 shows the comparison of the measured value of Fuji Heavy Industries and Toyota Motor Corporation. Fuji Heavy Industries and Toyota Motor

Table 10 Daihatsu and Fuji heavy industry

Date	Comparison of the measured value	Date	Comparison of the measured value
20080602	○	20080617	○
20080603	○	20080618	○
20080604	○	20080619	×
20080605	○	20080620	×
20080606	○	20080623	○
20080609	×	20080624	×
20080610	○	20080625	○
20080611	○	20080626	×
20080612	○	20080627	○
20080613	○	20080630	○
20080616	○		76.2 %

Table 11 Prediction result

Date	Toyota	Daihatsu	Both companies
20080602	–	–	–
20080603	–	○	○
20080604	–	×	×
20080605	○	○	○
20080606	–	×	×
20080609	–	○	○
20080610	○	○	○
20080611	×	×	×
20080612	–	×	×
20080613	○	–	○
20080616	–	○	○
20080617	–	–	–
20080618	○	○	○
20080619	×	×	×
20080620	○	–	○
20080623	–	○	○
20080624	–	○	○
20080625	×	–	×
20080626	–	○	○
20080627	○	–	○
20080630	×	×	×
Correct predictions (%)	60.0	60.0	63.2

Corporation showed the same movement was 81.0 %. Table 10 shows the comparison of the measured value of Fuji Heavy Industries and Daihatsu Motor Corporation. Fuji Heavy Industries and Daihatsu Motor Corporation indicated the same movement was 76.2 %.

Table 11 shows the predicted results. Prediction accuracy when using Toyota Motor stock values as related data was 60.0 %, using Daihatsu Motor stock values as related data was 60.0 %, using the stock prices of both as relevant data was the best, 63.2 %.

5 Conclusion

In this paper, we investigated knowledge acquisition about the market fluctuations in stock markets by using the rough set theory.

The intra-day trading data are the record of all the trading transactions related to all the stocks. We converted 1 min chart data to the intra-day trading data and acquired the knowledge in order to predict future changes by using the regression line and rough set. We considered whether we can predict the fluctuation movements in the market using the knowledge acquired, and compared the actual movements of stock prices with the model forecasts. In addition, we predicted the price movement using the proposed model. It was predicted by using the data in June 2008. The correct rate of predicted values in one month was 63.2 %. Error rate of the predicted value will be 36.8 %. Profit margin obtained by subtracting the error rate from the correct answer rate is 26.4 %. This is a very high value. We were able to visualize as a rule knowledge. This is an advantage over other approaches. By using the rough set, we were able to improve these results.

References

1. Matsumoto, Y., Watada, J.: Improvement of chaotic short-term forecasting on fuzzy reasoning and tuning on genetic algorithm. Jpn. Soc. J. Fuzzy Theory Intell. Inf. **16**(1), 44–52 (2004)
2. Pawlak, Z.: Rough sets. Int. J. Comput. Inf. Sci. **11**(5), 341–356 (1982)
3. Pawlak, Z.: Rough Sets—Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers (1991)
4. Tan, S., Cheng, X., Xu, H.: An efficient global optimization approach for rough set based dimensionality reduction. Int. J. Innov. Comput., Info. Control **3**(3), 725–736 (2007)
5. Goh, C., Law, R.: Incorporation the rough sets theory. Chemometr. Intell. Lab. Syst. **47**(1), 1–16 (2003)
6. Azibi, R., Vanderpooten, D.: Construction of rule-based assignment models. Eur. J. Oper. Res. **138**(2), 274–293 (2002)
7. Beynon, M.J., Peel, M.J.: Variable precision rough set theory and data discrimination: an application to corporate failure prediction. Omega **29**(6), 561–576 (2001)
8. Li, R., Wang, Z.O.: Mining classification rules using rough set and neural networks. Eur. J. Oper. Res. **157**(2), 439–448 (2004)
9. Quafafou, M.: α-RST: a generalization of rough set theory. Inf. Sci. **124**(4), 301–316 (2000)
10. Greco, S., Matarazzo, B., Slowinski, R.: Rough sets theory for multi-criteria decision analysis. Eur. J. Oper. Res. **129**(1), 1–47 (2001)

11. Jhieh, Y., Tzeng, G., Wang, F.: Rough set theory in analyzing the attributes of combination values for insurance market. Expert Syst. Appl. **32**(1), 56–64 (2007)
12. Harada, T., Tanaka, R.: Analysis of Specifications for web screen-design using rough sets. J. Adv. Comput. Intell. Intell. Info. **10**(5), 688–694 (2006)
13. Kim, D., Bang, S.Y.: IRIS data classification using tolerant rough sets. J. Adv. Comput. Intell. Intell. Info. 4(5) (2000)
14. Walczak, B., Massart, D.L.: Rough set theory. Chemom. Intell. Lab. **47**(1), 1–16 (1999)
15. Predki, B., Slowinski, R., Stefanowski, R., Wilk, S.z.: ROSE-software implementation of the rough set theory. In: Polkowski, L., Skowron, A. (eds.) Rough Set and Current Trends in Computing. Lecture Notes in Artificial Intelligence, Springer, Berlin, pp. 605–608, (1998)
16. Predki, B., Wilk, S.z.: Rough set based data exploration using ROSE system. In: Ras, Z.W, Skowron, A. (eds.) Foundations of Intelligent Systems. Lecture Notes in Artificial Intelligence, Poland, Warsaw: Springer, pp. 172–180, (1999)
17. Pawlak, Z.: Rough classification. Int. J. Hum.-Comput. Stud. **51**(15), 369–383 (1999)
18. Gronhaug, K., Gilly, M.C.: A transaction cost approach to consumer dissatisfaction and complaint action. J. Econ. Psychol. **12**(1), 165–183 (1991)
19. Lin, C.,Watada, J., Tzeng, G.: Rough sets theory and its application to management engineering. In: Proceedings, international symposium of management engineering, Kitakyushu, Japan, pp. 170–176, (2008)
20. Tanaka, H Tsumoto, S.: Rough sets and expert system, Math. Sci., pp. 76–83 (1994)
21. Mori, N., Tanaka, H., Inoue, K.: Rough sets and Kansei: knowledge acquisition and reasoning from Kansei data, (2004)

Deep Neural Network Modeling for Big Data Weather Forecasting

James N. K. Liu, Yanxing Hu, Yulin He, Pak Wai Chan and Lucas Lai

Abstract The coming of the big data era brings the opportunities to greatly improve the forecasting accuracy of weather phenomena. Specifically, weather change is quite a complex process that is affected by thousands of variables. In the traditional computational intelligence models, we have to select the features from variables according to some fundamental assumptions, thus the correctness of these assumptions may crucially affect the prediction accuracy. Meanwhile, the principle of big data is to let data speaking, which means, when the volume of data is big enough, the hidden statistical disciplines in domain data will be revealed by the data set itself. Therefore, if massive volume of weather data is employed, we may be able to avoid using assumptions in the models, and we have the opportunity to improve the weather prediction accepted by learning the correlations hidden in the data. In our investigation, we employ a new computational intelligence technology called stacked Auto-Encoder to simulate hourly weather data in 30 years. This method can automatically learn the features from massive volume of data set via layer-by-layer feature granulation, and the large size of the data set can make sure that the complex deep model does avoid the overfitting problem.

J. N. K. Liu (✉) · Y. Hu · L. Lai
Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong
e-mail: james.liu@polyu.edu.hk

Y. Hu
e-mail: csyhu@comp.polyu.edu.hk

L. Lai
e-mail: caskclai@yahoo.com

Y. He
College of Mathematics and Computer Science, Hebei University, Baoding, China
e-mail: yulinhe@ieee.org

P. W. Chan
Hong Kong Observatory, 134A Nathan Road, Kowloon, Hong Kong
e-mail: pwchan@hko.gov.hk

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_19,

The experimental results demonstrate that using the new represented features in the classical model can obtain higher accuracy in time series problems.

Keywords Weather forecasting · Big data · Deep Neural Network

1 Introduction

From the beginning of human's history, people never cease their efforts on predicting the trend of weather changes. Every step forward of weather forecasting technology has great academic and practical significance. This is not only because of the changes of climate that may greatly impact people's daily life, but also the fact that the advancing investigation of weather forecasting can reflect the progress of human's ability to know the earth.

Many significant research efforts are utilized to develop weather forecasting methods including computational intelligence technologies that have been accepted as appropriate means for weather forecasting and reported encouraging results since 1980s [6, 7, 17, 19, 21, 32]. However, the coming of the big data era brings the opportunities to improve the forecasting accuracy of weather phenomena in advance. Some conventional difficulties in weather forecasting tasks are expected to be solved with big data/large volume of weather information. Specifically, for weather forecasting tasks, the variation tendency of atmospheric phenomenon is quite unstable and complex, therefore, thousands of related variables are changing every second so that a small change of a certain variable may greatly affect the weather condition [16]. Unfortunately, the number of variables which can be handled in a certain model is limited. Especially, for computational intelligence models, if too many variables are employed, the overfitting problem is very difficult to be avoided with smaller number of training samples [3]. Accordingly, some fundamental assumptions are required, and the accuracy of the forecasting results highly depends on the correctness of initial condition of assumptions [15, 33].

The conception of "big data" refers to the increasing volume of the data sets that used to analyze problems in different research domains [37]. Combined with statistical methods and computational intelligence technologies, big data has brought a revolution to many traditional research fields including the meteorology, genomics, connectomics, complex physics simulations, and biological and environmental research [30], etc. The principle of big data is to "let data speaking", which means, when the volume of data is big enough, the hidden relevance in data set will be revealed via the statistical disciplines [1, 9, 31, 38]. Therefore, if massive weather data is employed, we may avoid using assumptions in our model, and we have the opportunity to directly analyze the correlations hidden in the weather data. In so doing, the generalization of the models and accuracy of the results are expected to be improved ultimately.

The true significance of the term “big data” not only concentrates on larger size of the data sets, but also refers to the suitable strategy to process the obtained large data set. Computational intelligence models, particularly Neural Networks (NNs), are good tools to discover the statistical rules hidden in the big data sets and have obtained some successful reputation in the previous big data applications [12, 34]. In computational intelligence field, a very prevalent conventional conception was that shallow and simple models, e.g. Support Vector Machines (SVMs) and Single Layer Feedforward Networks (SLFNs) can provide better performance than complex and deep models, e.g. NNs with deep (multi-layer) architectures in the big data environment [3, 35]. Meanwhile, previous NNs with multi-layer architecture have their own inferiorities including (1) huge computational complexity; (2) a complex NN model with too many parameters is inevitable to the overfitting problem. Nevertheless, the studies since 2006 undertaken by Hinton [13, 14] and followed by other researchers hold on the opposite conception: (1) NNs with deep structure may provide a superior learning capacity [3, 18]; (2) the newest proposed Deep Neural Network (DNN) approach, also well-known as Deep Learning (DL), employs a so-called layer−wise unsupervised pre−training mechanism to solve the training difficulties efficiently [5] and (3) particularly, in big data environment, despite the number of parameters in DNN is more than that of shallow models, the overfitting problem can also be avoided because of the huge amount of data samples [28].

Compared with simple and shallow models, NNs with deep architecture can provide a higher learning ability. Although the back-propagation NNs with three layers have been proved that can theoretically approximate any nonlinear functions with arbitrary precision [10], functions that can be compactly represented by a deep architecture might be required to handle an exponential number of computational elements (parameters) to be represented by a depth architecture. More precisely, functions that can be compactly represented by a depth k architecture might be requiring an exponential number of computational elements to be represented in a depth $k - 1$ architecture. Since the number of computational elements one can afford depends on the number of training examples available for tuning or selecting them, the consequences are not only computational but also statistical: poor generalization may be expected when using an insufficiently depth architecture for representing some functions [3, 5].

The core technologies in DNN is the layer−wise unsupervised pre−training mechanism, by such training method, the original information in the raw data can be represented [5] or granulated [24, 25]. By such granulation, the raw data in original feature space may be mapped into a new feature space, and the principle for such mapping is an information granula of interesting [2]. In a larger data environment, the significance of granulation becomes more important, we need some approaches for mining the knowledge in big data sets. With the granulation, the hidden relevance in the data set maybe extracted and represented layer by layer [5, 24]. Such a feature representation may greatly improve the performance of traditional computational intelligence models.

In our investigation, we apply a multi-layer model to predict the weather change in the next 24 hour with a big data set. The massive data involving millions

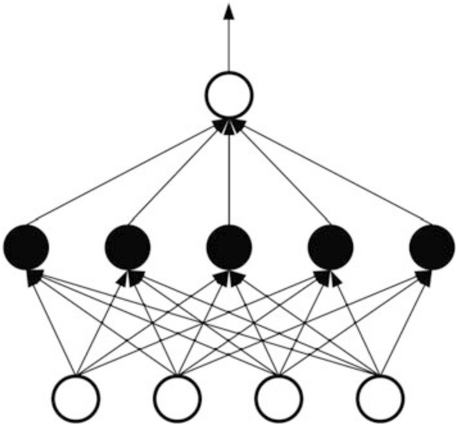


Fig. 1 A typical shallow feedforward network with one hidden layer

of weather records is provided by The Hong Kong Observatory (HKO).[1] Our training method is to use the latest proposed greedy layer-wise unsupervised pre-training algorithm followed by a supervised fine-tuning. In detail, we choose a revised autoencoder algorithm to build the network, the DNN is used to learn the features from the larger volume of raw data, and we will evaluate the learned features according to the prediction accuracy. The contribution and significance of our investigation demonstrates that: compared with the classical models, NN with Deep architectures can improve the prediction accuracy in weather forecasting field; moreover, the positive results show the potential of DNN model in big data; last but not least, DNN has won some encouraging results in research field including Computer Vision [14], Speech Recognition [23], Natural Linguistic Programming [4] and Bioinformation, our investigation will show that DNN also has great potential in time series problems, especially in weather forecasting domain [22].

This chapter is organized as follows: in Sect. 2 we will introduce some background knowledge, mainly including how to train a DNN model layer by layer. Section 3 briefly discusses the DNN model for weather data simulation. Section 4 discusses the experiments along with some comparative analysis. The last section gives the conclusion and future work.

2 Background Knowledge

In this section, some background knowledge is presented. Specifically, we mainly introduce the greedy layer−wise unsupervised pre−training approach and the stacked Auto-Encoder based DNN.

[1] http://www.hko.gov.hk/contente.htm.

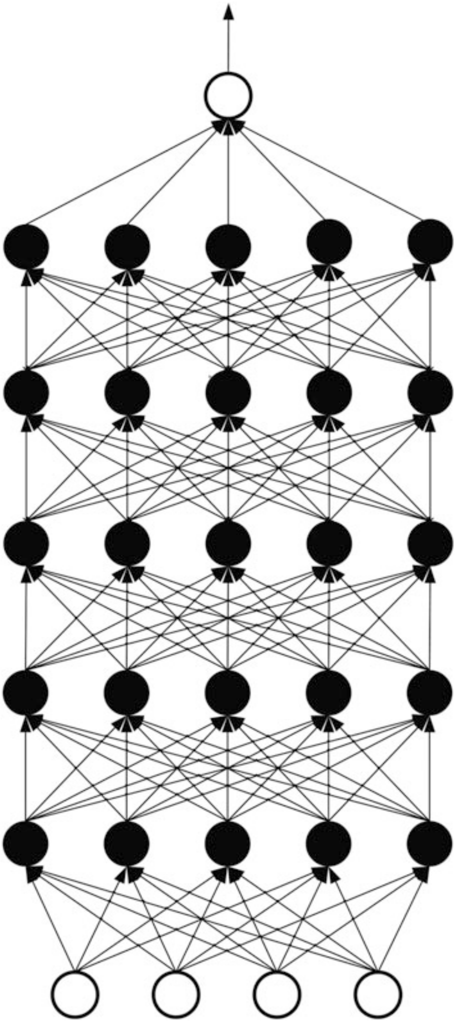


Fig. 2 A traditional NN with deep architecture (simply adding extra hidden layers to the shallow model), that shows complex structure, hard to train, and easily overfitting

2.1 Greedy Layer-Wise Unsupervised Pre-training: Auto−encoder Granulation

The essential challenge in training a NN with deep architecture is to deal with the strong dependencies that exist during training between the parameters across layers [11]. In previous investigations, researchers found that simply adding layers to a classical shallow Feedforward Network cannot overwhelm the mentioned challenge. Figure 1 shows the architecture of the classical SLFNs and Fig. 2 gives the earlier model of NN with multi-layer architecture.

Training deep architectures involves a potentially intractable non-convex optimization problem, and there were no inadequate algorithms for training fully-connected deep architectures until Hinton et al. introduced a learning algorithm that greedily trains one layer at a time in 2006 [14]. Shortly after, strategies for building deep architectures from related variants were proposed by Bengio [22] and Ranzato [29]. They solved the training problem of DNN in two phases: for the first phase, unsupervised pre-training, all layers are initialized using this layer-wise unsupervised learning signal; for the second phase, fine-tuning, a global training criterion (a prediction error, using labels in the case of a supervised task) is minimized. Such training approach is called the greedy layer−wise unsupervised pre−training, and DNN training with this mechanism since then has been applied with success in many fields, which is widely known as the term "Deep Learning".

Several unsupervised training models have been proposed and investigated since 2006. These models are categorized into the family of greedy layer−wise unsupervised pre−training approaches and employed to build the deep architecture of NN, e.g. Restricted Boltzmann Machines (RBMs), Stacked Auto-Encoder, CNNs, etc. According to Andrew NG in 2007, the selection of different greedy layer-wise unsupervised pre-training approaches in DNN gives little effect to the final result [8]. Therefore, with the consideration of the attribute type of the weather data, i.e., the collected data are all real numbers, in this investigation, we choose the Stacked Auto-Encoder to build the deep architecture of our NN model.

The Stacked Auto-Encoder, as its name suggests, is a stacked architecture NN that applies Auto-Encoder in each layer. In computational intelligence field, NN means a network of neurons with different architectures (e.g., NN in Figs. 1 and 2). A single "neuron" is a computational unit that taken as input vector $X = x_1, x_2, \ldots, x_n$ (and a "+1" intercept term), and outputs $h_{W,b}(x) = f(W^T x) = f(\sum_{i=1}^{3} W_i x_i + b)$, where $f : \Re \mapsto \Re$ is called the activation function, and W is the weight matrix that stands for the connection among different neurons in the network. In most of cases, sigmoid function $f(z) = \frac{1}{1+\exp(-z)}$ is chosen as the activation function. A typical Auto-Encoder tries to learn a function $h_{w,b}(x) \approx x$. In other words, it is trying to learn an approximation to the identity function, so as to output $\hat{x}$ that is similar to x. The identity function seems a typically trivial function trying to learn; but by placing constraints on the network, such as by limiting the number of hidden neurons, we can discover interesting structure about the data [29], e.g., for a data set, suppose that the original samples are collected from a 100-dimensional feature space, i.e. $x \in \Re^{100}$, set that there are 50 hidden neurons in the hidden layer, based on the requirement $h_{w,b}(x) \approx x$, the network is forced to learn a compressed representation of the input. That is, given only the vector of hidden unit activations $a^{(2)} \in \Re^{50}$, it must try to reconstruct the 100-dimensional input x. An illustration of Auto-Encoder is shown in Fig. 3. If the inputs were completely random, each x_i comes from an I.I.D. Gaussian independent of the other features, then this compression task would be very difficult. But if there is a certain structure hidden in the data, for example, if some of the input features are correlated, such as

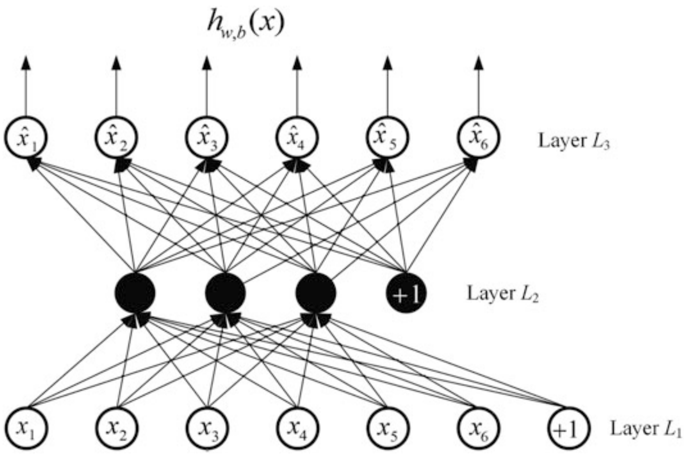


Fig. 3 An illustration of auto-encoder algorithms. Layer L_1 is the input layer, and L_3 is the output layer. Via hidden layer L_2, we hope to represent the information x in layer L_1, so that the output $\hat{x}$ in L_3 can approximate the raw data x

in the feature space of time series analysis, then this algorithm will be able to discover some of those correlations.

The loss function of Auto-Encoder is:

$$J(W,b) = \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{W,b}\left(x^{(i)}\right) - x^{(i)}\right\|^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ji}^{(l)}\right)^2 \tag{1}$$

where m is the number of training samples. The objective of the Auto-Encoder is to minimize Eq. (1) in order to make sure that the output $h_{W,b}(x^{(i)})$ can approximate the raw data $x^{(i)}$ as far as possible. The second term in Eq. (1) is a regularization term (also called a weight decay term) controlled by the weight decay parameter λ that tends to decrease the magnitude of the weights, and helps prevent overfitting. We can minimize Eq. (1) by *gradient descent* to compute the configuration of the network.

In some special cases, the number of hidden neurons is large (perhaps even greater than the number of dimensions of input vectors), we can still discover interesting structure by imposing other constraints on the network. In particular, if we impose a **sparsity** constraint on the hidden neurons, then the autoencoder will still discover interesting structure in the data, even if the number of hidden neurons is large. To achieve this, we would like to define a ρ as sparsity parameter, typically a small value close to zero. In other words, if we use $a_j(x)$ to denote the activation of the jth hidden neuron when the network is given a specific input x, we hope that the average activation $\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}\left[a_j(x^{(i)})\right]$ of each hidden neuron to be close to ρ. Then a revised loss function is employed as:

$$J_{\text{sparse}}(W,b) = J(W,b) + \beta \sum_{j=1}^{n} \mathrm{KL}\left(\rho \| \hat{\rho}_j\right), \tag{2}$$

where n is the number of the hidden neurons, $J(W,b)$ is defined in Eq. (1), and β controls the weight of the sparsity penalty term. The term $\hat{\rho}_j$ (implicitly) depends on W, b, and the last term $\mathrm{KL}(\rho \| \hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j}$ is the Kullback-Leibler (KL) divergence between a Bernoulli random variable with mean ρ and a Bernoulli random variable with mean $\hat{\rho}_j$.

2.2 The Deep Neural Networks and Layer by Layer Granulations

In a stacked Auto-Encoder based DNN, for each layer, we use an Auto-encoder to train the parameters in this layer, and then combined these layers together. Specifically, in the training process of each layer, as shown in Fig. 3, the input vectors have to pass three layers, and the vectors in hidden layers (layer L_2, and for simplicity, we call the vectors in layer L_2 as the transformed vectors of the initially input vectors) are representations of the input vectors and can be used to reconstruct the input vectors. Thus, in every layer of the DNN, the input of the current layer is the output of the previous layer, then we train the input data via an Auto-Encoder, and use the transformed vectors as the output of the current layer. Figure 4 shows the detailed mechanism of stacked Auto-Encoder based DNN.

Observing the NN in Fig. 4, we can find that the principle of layer-wise unsupervised pre-training based DNN is to map the raw data into a new feature space layer by layer. For computational intelligence model, the selection of features can greatly impact the accuracy of models. Therefore, how to select features is one of the core and universal problems. The DNN provides us a method to learn features instead of manually selection like we did previously. In DNN, data in raw feature space can be mapped into a new space and regularized in each layer. Thus, in each layer, input features can be reconstructed and new features can be generated; finally, these generated features can be applied by the model in the top layer. In big data environment, the requirement of learning feature is more important, and the larger size of the data can improve the quality of the generated features and guarantee the avoidance of overfitting.

The DNN based on stacked Auto-Encoder can be also seen as a branch or an extension of Granular Computing(GC) [26]. Information granules can be regarded as collections of objects that exhibit some similarity in terms of their properties or functional appearance. Actually, early in 2001, Pedrycz has already applied NN model to process granulation problem [26]. Generally, information granules defined in some space X can be treated as a mapping:

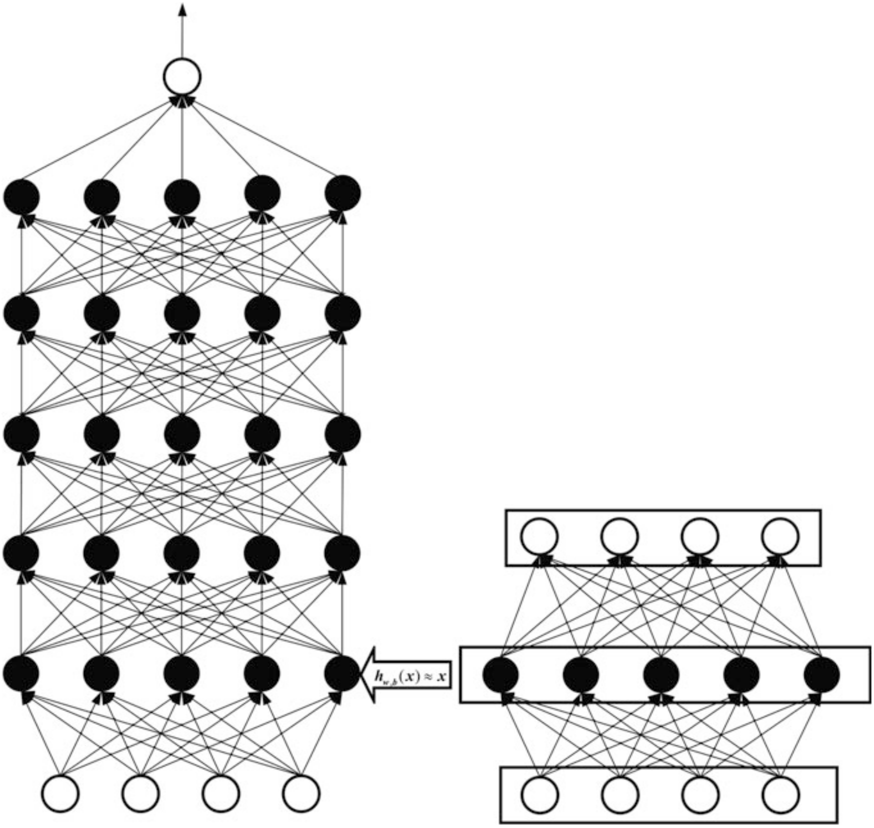


Fig. 4 A DNN with Stacked Auto-Encoder method, by which each layer is greedily pre-trained with an unsupervised Auto-Encoder algorithm and to learn a nonlinear transformation of its input (the output of the previous layer) that captures the main variations in its input, i.e. $h_{W,b}(x) \approx x$

$$A : X \rightarrow \zeta(x) \tag{3}$$

where A is an information granule of interest, ζ denotes a formal framework of information granules. From this definition, we can see that the granulation of a universe involves grouping of similar elements into granules to form coarse-grained views of the universe [39]. The only difference between classical GC and the Auto-Encoder (or layer-wise unsupervised pre-training algorithms) is that, GC is to group and reconstruct samples, but Auto-Encoder is to group and reconstruct the feature of the samples, therefore, Deep NN based on stacked Auto-Encoder can be considered as a branch or an extension of GC, which granulates the information of the raw data layer by layer.

2.3 Support Vector Regression

The principle of layer-wise unsupervised pre-training based DNN is to learn the features, therefore, we have to choose the model applied in the top layer, or output layer to process the learned features. In our experiment, SVM is employed.

Based on the Structural Risk Minimization (SRM) principle, SVM method seeks to minimize an upper bound of generalization error instead of the empirical error as in other NNs. Additionally, SVM models generate the regression function by applying a set of high-dimensional linear functions. The Support Vector Regression (SVR) function is formulated as follows:

$$y = w\phi(x) + b \tag{4}$$

where $\phi(x)$ is called the feature, which is nonlinear and mapped from the input space $\Re^n$. y is the target output value we want to estimate. The coefficients w and b are estimated by minimizing:

$$R = \frac{1}{2}\|w\|^2 + \frac{1}{n}C\sum_{i=1}^{n} L_\varepsilon(d_i, y_i) \tag{5}$$

where:

$$L_\varepsilon(d, y) = \begin{cases} |d - y| - \varepsilon, |d - y| \geq \varepsilon \\ 0, \text{otherwise} \end{cases} \tag{6}$$

Equation (5) is the risk function consisting of the empirical error and a regularization term that is derived from the SRM principle. The term $\frac{1}{n}\sum_{i=1}^{n} L_\varepsilon(d_i, y_i)$ in Eq. (5) is the empirical error (risk) measured by the ε-insensitive loss function (ε-insensitive tube); in the meanwhile, the term $\frac{1}{2}\|w\|^2$ is the regularization term. The constant $C > 0$ is taken as the regularized constant that determines the trade-off between the empirical error (risk) and the regularization term. Increasing the value of C will add importance to the empirical risk in the risk function. ε is called the tube size of the loss function and it is equivalent to the accuracy approximation placed on the training data points. Both C and ε are user-prescribed parameters.

Then the slack variables ζ and ζ^* which represent the distance from the real values to the corresponding boundary values of ε-insensitive tube are introduced. With these slack variables, Eq. (5) can be transformed to the following constraint based optimization:

Minimize:

$$R(w, \zeta, \zeta^*) = \frac{1}{2}ww^T + C^*\left(\sum_{i=1}^{n}(\zeta + \zeta^*)\right) \tag{7}$$

Subject to:

$$\begin{aligned} & w\phi(x_i) + b_i - d_i \leq \varepsilon + \zeta_i^* \\ & d_i - w\phi(x_i) - b_i \leq \varepsilon + \zeta_i \\ & \zeta_i, \zeta_i^* \geq 0, \quad i = 1, 2, \ldots, n \end{aligned} \tag{8}$$

Finally, by introducing the Lagrangian multipliers and maximizing the dual function of Eq. (4), it can be changed to the following form:

$$\begin{aligned} R(\alpha_i - \alpha_i^*) = & \sum_{i=1}^{n} d_i(\alpha_i - \alpha_i^*) - \varepsilon \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) \\ & - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i - \alpha_i^*) \times \left(\alpha_j - \alpha_j^*\right) (\Phi(x_i) \cdot \Phi(x_k)) \end{aligned} \tag{9}$$

with the constraints:

$$\sum_{j=1}^{n} (\alpha_i - \alpha_i^*) = 0, \quad 0 \leq \alpha_i \leq C, \quad 0 \leq \alpha_i^* \leq C, \quad i = 1, 2, \ldots, n \tag{10}$$

where α_i and α_i^* are Lagrangian multipliers which satisfy $\alpha_i \times \alpha_i^* = 0$, the general form of the regression estimation function can be written as:

$$f(x, \alpha_i, \alpha_i^*) = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) K(x, x_i) + b \tag{11}$$

$K(x_i \cdot x)$ is called the kernel function. It is a symmetric function $K(x_i \cdot x) = (\Phi(x_i) \cdot \Phi(x))$ satisfying Mercer's conditions. When the given problem is a nonlinear problem in the primal space, we may map the sample points into a high-dimensional feature space where the linear problem can be performed. Linear, Polynomial, Radial Basis Function (RBF) and Sigmoid are four main kernel functions in use. As we discussed above, in most of the time series forecasting problems, the SVR employs RBF kernel function to estimate the nonlinear behavior of the forecasting data set because RBF kernels tend to give good performance under general smoothness assumptions.

3 Weather Prediction with Deep Neural Networks

The main task of our investigation is to employ DNN based on stacked Auto-Encoder to predict weather informations, more specifically, we hope to predict two kinds of weather information, temperatures and wind speed, in the next few hour. Univariate time series regression is the most fundamental and most widely applied model in short-term weather forecasting. Generally speaking, for a certain variable, the objective of univariate time series regression is to find the relationship between its status in a certain future time point and its status in a series of past time points, and estimate its future status via:

$$v_t = f(v_{t-1}, v_{t-2}, \ldots, v_{t-n}) \tag{12}$$

To obtain f, earlier investigations employ models such as Linear Regression, Generalized Linear Model, Autoregressive Integrated Moving Average Mode, etc. After the computational intelligence technologies become a hot research field, investigators found that some intelligence technologies models, such as NNs, SVMs, and fuzzy models could provide higher generalization on some complex, nonlinear, and unstable domains including weather forecasting tasks.

Weather data has some particularities. More specifically, there is season-to-season, and year-to-year variability in the trend of weather data. The cycle could be multi-month, multi-season or multi-year, and the main difficulty of investigations is to capture all the possible cycles.

In our investigation, we will simply input the data sets into our model. The architecture of the applied model is as the NN in Fig. 3. The input n-dimensional vector is composed by the status in $(t - 1)$th, $(t - 2)$th,…, $(t - n)$th time points, we try to use the DNN to represent these status, and employ a SVR to estimate the status in tth time point. Since our data set is quite large, we hope the seasonal cycles can be captured via the massive volume of data by the superior learning ability of the multi-layer structured NN.

4 Experimental Results and Analysis

From the discussion in previous sections, we can see that the key point of employing a DNN is to learn the features, or granulate raw data into a new feature space via a multi-layer NN. Therefore, our experiments concentrate on the evaluation of the learned features: two types of weather data, with very large data sets, are employed and simulated, and the comparison of results is conducted between models using raw features and models with represented features.

4.1 Data Collection

The HKO has provided great support to our investigation. Based on our collaboration with HKO, a massive volume of high quality real weather data could be applied in our experiment. Two types of historical weather data sets, the wind speed data and temperature data are employed in our model. The time range of the data sets is almost 30 year long, which covers the period from January 1, 1983 to

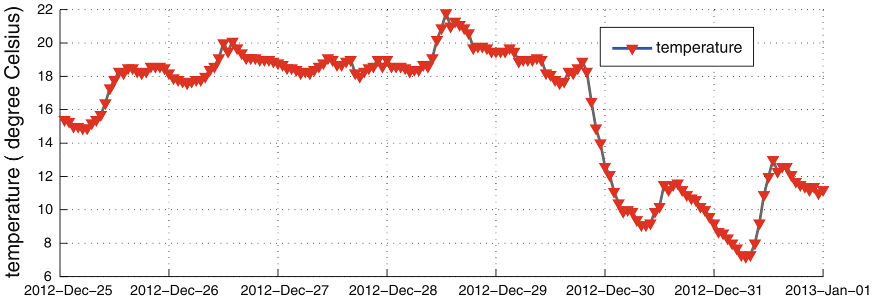


Fig. 5 The distribution of temperature data in the last week of the data set

December 31, 2012. In detail, the number of temperature records is more than 260,000, and the total number of records in the wind speed experiment is more than 1,560,000.

Compared with the temperature data which has only one dimension (measured in degree Celsius), the wind speed data has two dimensions: the polar coordinate for the wind direction (measured in degree angle) and the speed (measured in meters per second). Moreover, in the raw data set, for a certain time points, the direction of the air motion is not stable, i.e. the wind direction at that time point is not fixed. Such condition is denoted as ''variable'' in the raw data. Therefore, according to the requirement of our algorithm, we have to do some pre-processing on the data sets.

4.2 Data Pre-processing

Compared with the temperature data which is a scalar quantity only having one dimension (as Fig. 5), the wind speed data (in a fixed horizontal plane) is a vector quantity that has two dimensions in the polar coordinate (as Fig. 6), i.e. the angle to show its direction and the speed to measure the velocity in this direction: the polar coordinate and the speed [27]. However, since our model focuses on univariate time series problems, we have to transform the data set to satisfy the model's requirement. According to the physical significance of the two dimensions, we denote the angle as θ and the speed as v to obtain:

$$v^0 = \cos\theta \cdot v \tag{13}$$

where v^0 is the vector components of the wind speed in 0° angle direction (as Fig. 7). Thus, what we actually simulate is the time series of the speed component of the air motion in 0 degree angle direction. Moreover, there are about 3 % wind speed data with the direction valued as "variable", for such condition, we consider it as a missing value in the data set and use the average value of the wind direction in its previous time point and its next time point to replace the value "variable".

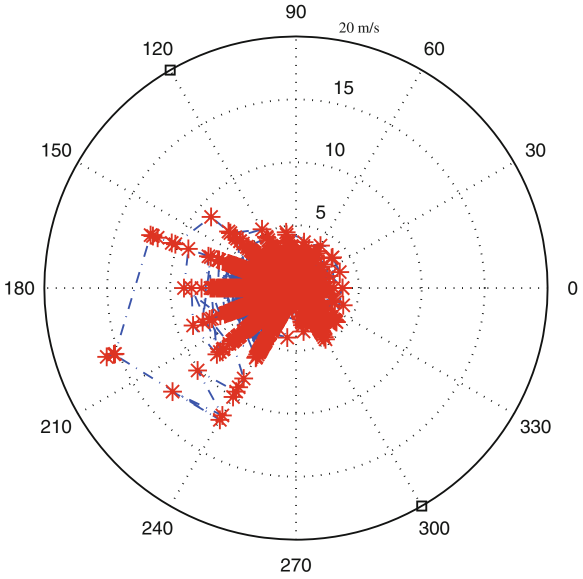


Fig. 6 The distribution of wind speed data in polar coordinate

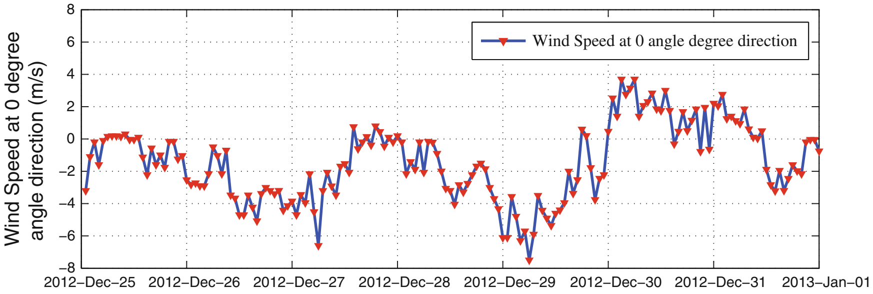


Fig. 7 The distribution of wind speed at a fixed direction in the last week of the data set

4.3 Evaluation Criteria

Three criteria are applied in our investigation to evaluate the prediction performance of the used models.

Normalized Mean Squared Error (NMSE) measures the deviation between the actual values and the predicted values. The smaller the values are, the closer the predicted values to the actual values. The formula of NMSE is:

$$\mathrm{NMSE} = 1/\left(\delta^2 n\right)\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \tag{14}$$

where

$$\delta^2 = 1/(\mathrm{n}-1)\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \tag{15}$$

Directional symmetry (DS) indicates the percentage correctness of the predicted direction. The formula of DS is:

$$\mathrm{DS} = (100/n) \times \sum_{i=1}^{n} d_i \tag{16}$$

where

$$d_i = \begin{cases} 1, (x_i - x_{i-1})(\hat{x}_i - \hat{x}_{i-1}) \geq 0 \\ 0, \text{otherwise} \end{cases} \tag{17}$$

We also employ R^2 value to evaluate the whole simulation ability of our model, the R^2 indicates how well our model can explain the raw data set. The R^2 value of a model is:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{18}$$

where $SS_{tot} = \sum_{i=1}^{n} (\hat{x}_i - x_i)^2$ and $SS_{tot} = \sum_{i=1}^{n} (\hat{x}_i - \bar{x})^2$.

4.4 Experimental Results and Discussion

In our first experiment, we use a 4-layer DNN model to predict the temperature in the next time point. More specifically, we tried to use the 7-day hourly temperature data to forecast the temperature in the next 24 hour, The NN model is with a single input layer, two stack-organized Auto-Encoder layers, and the top layer which used SVR to output the prediction results. In this experiment, since the sparsity is less considered in weather forecasting, we adjusted the value of β to a relatively small value, and set the number of hidden nodes as 84. The experiments were conducted in a CPU cluster with 6 Intel i7 processors with MatLab 2012a. The experiment is based on the 10-fold cross-validation, thus in each cycle, the training set has more than 230,000 randomly selected records, and about 26,000 samples are selected as training set. The result is compared with classical SVR. Note that the parameter in the classical SVR is set as same as in the top SVR layer of our model. Table 1 and Fig. 8 give the result.

From Fig. 8 we can observe that both SVR and DNN can simulate the real data very well after training with a big data set, the predicted results are almost coincide

Table 1 The comparison of temperature prediction by SVR and DNN

Model	NMSE	DS	R^2
Classical SVR	2.179e−2	0.75	0.872
DNN with SVR in top layer	8.117e−3	0.79	0.915

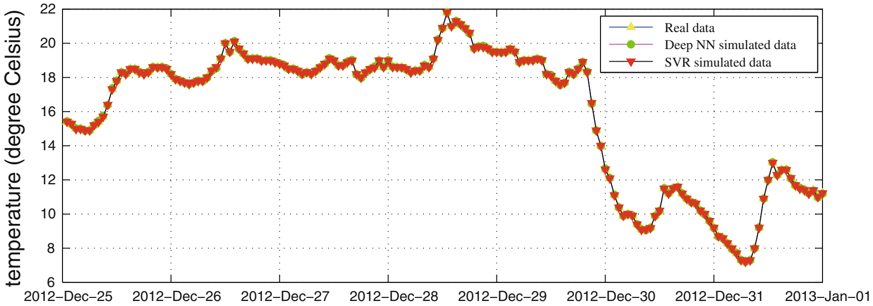


Fig. 8 The results of temperature prediction for the date in the last week

with the real data. Such an encouraging result demonstrates the advantage of big data for univariate time series problem: the massive volume of the data set can help us to approximate the statistical discipline with a higher accuracy.

Results shown in Table 1 demonstrate the positive role of the DNN in the temperature simulation task. As we discussed in above, both of the two models have the same configuration in SVR part, the only difference is that in the DNN, we represent/granulate the raw features, and use the new features to train the SVR in the top layer. We can see that the DNN model greatly reduces the NMSE (the NMSE in SVR model has already been very small, but after the feature representation/granulation, the NMSE becomes even smaller), and the higher R^2 value also proved that, with the represented features via a DNN, the SVR model in the top layer can learn the raw data much better.

From Figs. 5 and 7, we may observe that compared with wind speed data, the fluctuation of temperature data (in short time) is much more smooth. Therefore, temperature simulation is not a challenging task. actually, in the training process of our experiment, the classical SVR model takes more time than DNN. This phenomenon may be attributed to two aspects: (1) the temperature data is very smooth so that the convergence of the NN is very quick [4]; (2) the represented features can reveal the principle of the given data set and consequently improve the performance of SVR [20, 36]. Therefore, the shortening of the training time gives two things: (1) the new feature space that reconstructed via DNN has positive effect for time series tasks; (2) the significance of our positive results obtained in temperature data set is limited.

In the second experiment, we change the data set. The wind speed data is employed. Since the change of wind speed data is much more unstable, the

Table 2 The comparison of wind speed prediction by SVR and DNN

Model	NMSE	DS	R^2
Classical SVR	0.3721	0.72	0.851
DNN with SVR in top layer	0.2522	0.83	0.871

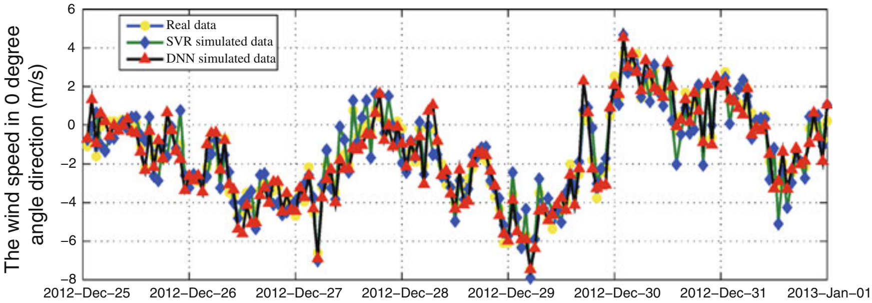


Fig. 9 The results of temperature prediction for the date in the last week

simulation of wind speed data is more difficult and has more academic and practical significance. We made some modifications on our model in the second experiment: we added two Auto-Encoder layers in the model in order to improve the learning ability of the network. The results are shown in Table 2 and Fig. 9.

Figure 9 shows the simulation results of the wind speed data in the last week. From Fig. 9, we can observe that after training with a big volume of wind speed data, the model can capture the main trend of changes, and the DNN can give a better performance than simply using SVR. Inspecting the criteria in Table 2, DNN can return a lower NMSE and higher R^2 value. Note that the DS value is greatly improved when the data is simulated with the DNN model, this maybe caused by the fact that features generated via DNN may have the largest possible variation, and such fact shows that the principle of DNN may be considered as an advanced form of Principal Component Analysis (PCA).

Our experiments only make comparison between Classical SVR and Stacked Auto-Encoder DNN with SVR in the top layer. Actually, some other models can also be applied to deal with weather data related time series problem. However, the main objective of our investigation is to attest the models' performance with the new represented/granulated features. The results demonstrate that compared with the raw features, the obtained features can explain the principle of the raw data set better. What is more, the DNN can be combined with many other models, and the obtained features can be employed to improve the performances of most models in computational intelligence field.

5 Conclusion and Future Work

Big data may bring revolutions to many research fields including weather forecasting. In this chapter, we explore an approach that using computational intelligence technologies to process massive volume of data. The proposed DNN model may granulate the features of the raw weather data layer by layer, and experimental results show that the new obtained features can improve the performances of classical computational intelligence models.

The contribution of our investigation is significant: we give an approach that using computational intelligence method to learn features with big data, and our experiments demonstrate that the DNN algorithm also has the potential to address time series problems.

The main future work of our investigation is that, we will try to employ our model on some more difficult weather data, such as rain fall data set; and moreover, we will continue exploring the theoretical principle of computational intelligence, especially, we will try to give the mathematical explanation of the DNN.

Acknowledgment The authors would like to acknowledge the partial support of the CRG grants G-YL14, G-YM07 of The Hong Kong Polytechnic University.

References

1. Aronova, E., Baker, K.S., Oreskes, N.: Big science and big data in biology: from the international geophysical year through the international biological program to the long term ecological research (LTER) network, 1957–present. Hist. Stud. Nat. Sci. **40**(2), 183–224 (2010)
2. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Springer, Berlin (2003)
3. Bengio, Y.: Learning deep architectures for AI, vol. 2. Now Publishers Inc. (2009)
4. Bengio, Y.: Deep learning of representations for unsupervised and transfer learning. J. Mach. Learn. Res. Proc. Track **27**, 17–36 (2012)
5. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. Adv. Neural Inf. Process. Syst. **19**, 153 (2007)
6. Chen, S.M., Hwang, J.R.: Temperature prediction using fuzzy time series. IEEE Trans. Syst. Man Cybern. B: Cybern. **30**(2), 263–275 (2000)
7. Chen, S.M., Tanuwijaya, K.: Multivariate fuzzy forecasting based on fuzzy time series and automatic clustering techniques. Expert Syst. Appl. **38**(8), 10594–10605 (2011)
8. Coates, A., Ng, A.Y., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: International Conference on Artificial Intelligence and Statistics, pp. 215–223 (2011)
9. Condie, T., Mineiro, P., Polyzotis, N., Weimer, M.: Machine learning for big data. In: Proceedings of the 2013 International conference on Management of Data, pp. 939–942. ACM (2013)
10. Cybenko, G.: Approximation by superpositions of a sigmoidal function. Math. Control Signals Syst. **2**(4), 303–314 (1989)
11. Erhan, D., Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P., Bengio, S.: Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. **11**, 625–660 (2010)

12. Herrero, J., Valencia, A., Dopazo, J.: A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics **17**(2), 126–136 (2001)
13. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)
14. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
15. Kissinger, C.R., Gehlhaar, D.K., Fogel, D.B.: Rapid automated molecular replacement by evolutionary search. Acta Crystallogr. D Biol. Crystallogr. **55**(2), 484–491 (1999)
16. Kuligowski, R.J., Barros, A.P.: Localized precipitation forecasts from a numerical weather prediction model using artificial neural networks. Weather Forecast. **13**(4), 1194–1204 (1998)
17. Kwong, K., Wong, M.H., Liu, J.N., Chan, P.: An artificial neural network with chaotic oscillator for wind shear alerting. J. Atmos. Oceanic Technol. **29**(10), 1518–1531 (2012)
18. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 609–616. ACM (2009)
19. Liu, J., Kwong, K.M., Chan, P.W.: Chaotic oscillatory-based neural network for wind shear and turbulence forecast with lidar data. IEEE Trans. Syst. Man Cybern. C: Appl. Rev. **42**(6), 1412–1423 (2012)
20. Liu, J.N., Hu, Y.: Application of feature-weighted support vector regression using grey correlation degree to stock price forecasting. Neural Comput. Appl., 1–10 (2013)
21. Maier, H.R., Jain, A., Dandy, G.C., Sudheer, K.P.: Methods used for the development of neural networks for the prediction of water resource variables in river systems: current status and future directions. Environ. Model Softw. **25**(8), 891–909 (2010)
22. Miller, S.M., Geng, Y., Zheng, R.Z., Dewald, A.: Presentation of complex medical information: Interaction between concept maps and spatial ability on deep learning. Int. J. Cyber Behav. Psychol. Learn. (IJCBPL) **2**(1), 42–53 (2012)
23. Mohamed, A.R., Dahl, G.E., Hinton, G.: Acoustic modeling using deep belief networks. IEEE Trans. Audio Speech Lang. Process. **20**(1), 14–22 (2012)
24. Pedrycz, W., Bargiela, A.: Granular clustering: a granular signature of data. IEEE Trans. Syst. Man Cybern. B: Cybern. **32**(2), 212–224 (2002)
25. Pedrycz, W., Skowron, A., Kreinovich, V.: Handbook of Granular Computing. Wiley, New York (2008)
26. Pedrycz, W., Vukovich, G.: Granular neural networks. Neurocomputing **36**(1), 205–224 (2001)
27. Pielke, R.A.: Mesoscale Meteorological Modeling. Academic Press, US (2002)
28. Raina, R., Madhavan, A., Ng, A.Y.: Large-scale deep unsupervised learning using graphics processors. ICML **9**, 873–880 (2009)
29. Ranzato, M., Huang, F.J., Boureau, Y.L., Lecun, Y.: Unsupervised learning of invariant feature hierarchies with applications to object recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07, pp. 1–8. IEEE (2007)
30. Reichman, O., Jones, M.B., Schildhauer, M.P.: Challenges and opportunities of open data in ecology. Science (Washington) **331**(6018), 703–705 (2011)
31. Rushton, J.P., Irwing, P.: A general factor of personality (GFP) from two meta-analyses of the big five: and. Personality Individ. Differ. **45**(7), 679–683 (2008)
32. Sánchez Reinoso, C., Cutrera, M., Battioni, M., Milone, D., Buitrago, R.: Photovoltaic generation model as a function of weather variables using artificial intelligence techniques. Int. J. Hydrogen Energy **37**(19), 14781–14785 (2012)
33. Sundqvist, H., Berge, E., Kristjánsson, J.E.: Condensation and cloud parameterization studies with a mesoscale numerical weather prediction model. Mon. Weather Rev. **117**(8), 1641–1657 (1989)
34. Suzuki, K., Zhang, J., Xu, J.: Massive-training artificial neural network coupled with laplacian-eigenfunction-based dimensionality reduction for computer-aided detection of polyps in CT colonography. IEEE Trans. Med. Imaging **29**(11), 1907–1917 (2010)

35. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 3360–3367. IEEE (2010)
36. Wang, X., He, Q.: Enhancing generalization capability of SVM classifiers with feature weight adjustment. In: Knowledge-Based Intelligent Information and Engineering Systems, pp. 1037–1043. Springer, Berlin (2004)
37. White, T.: Hadoop: The Definitive Guide. O'Reilly (2012)
38. Wu, X., Zhu, X., Wu, G.Q., Ding, W.: Data mining with big data. IEEE Trans. Knowl. Data Eng. **26**(1), 97–107 (2014)
39. Yao, Y.: Information granulation and rough set approximation. Int. J. Intel. Syst. **16**(1), 87–104 (2001)

Current Knowledge and Future Challenge for Visibility Forecasting by Computational Intelligence

Wang-Kun Chen and Chung-Shin Yuan

Abstract Visibility forecasting, a challenging task of big data, is a complicated work in environmental simulation. There are many factors affecting the visibility change, such as humidity, cloud, and particulate concentration. The traditional statistical analysis cannot predict the visibility change very well, so the researchers try to apply various new methods. In this study, a review of the results of current knowledge, as well as future perspective about the forecasting of visibility, is presented. Because the environmental visibility is a typical type of big data, summarizing the meaningful information is benefit to the work of forecasting. Development of information granularity and computational intelligence help us to solve these problems. Various algorithms, such as grey theory, fuzzy theory, and neural network etc. provide superior capability in forecasting. However, how to effectively use the observed data is another great challenge. Finding scalable computational intelligence algorithms for visibility forecasting is essential. The knowledge from atmospheric dynamics and aerosol science helps us to explain the information derived from the big data of visibility. To spend more efforts in the interpretation of their relationship and develop more advanced mining methodologies helps us to construct the future model of visibility forecasting.

Keywords Forecasting model · Atmospheric visibility · Computational intelligence · Big data

W.-K. Chen (✉)
Department of Environment and Property Management, Jinwen University of Science and Technology, New Taipei, Taiwan
e-mail: wangkun@just.edu.tw

C.-S. Yuan
Institute of Environmental Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan
e-mail: ycsngi@mail.nsysu.edu.tw

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_20,

1 Introduction

1.1 *Knowing Visibility Forecasting from Physical and Chemical Point of View*

Visibility is an important physical and chemical phenomenon for environmental research. Because of this characteristic, so there are numerous studies from the perspective of atmospheric physics and chemistry in the past [7, 13, 14, 18] (Yuan et al. 2000). However, owing to the nature of this phenomenon is too complicated, so there are still many bottlenecks in grasping the visibility forecasting. For example, a number of parameters were introduced in order to more clearly explain the changes in visibility occurrence; nevertheless it also brings more uncertainty with the more parameters [4, 34]. As a result, it is worthy of our attention to develop a more efficient method to predict atmospheric visibility.

There are many factors which affect visibility forecasting. Yuan et al. [38] has studied the correlation of visibility with the chemical composition. Lee et al. [17] find the aerosol species has the influence to the visibility in Kaohsiung area. The size distribution are also found to have the effect to the atmospheric visibility (Yuan et al. 2000). The urban air quality is also an indicator of visibility (Yuan et al. 2000) [44].

Visibility can be used as an indicator of environmental quality. Wong has suggested that visibility can be used as indicator of visual senses, and the value of visibility can be regarded as the supplementary indicators of air quality [35, 36, 37]. Yuan has studied the influence of dust storm from mainland China to the characteristic of suspended particulate in Penghu area [39, 40]. The results reveals that visibility is closely correlated to both air pollutants and meteorological condition. Yuan has also used the artificial neural network to predict the concentration of PM_{10} and atmospheric visibility. The modeling results showed that the optimum input variables included the PM_{10} concentration, atmospheric pressure, surface radiation, relative humidity, atmospheric temperature, and cloud condition [41]. The dust storm from China is also an important factor of particulate concentration.

The paradigm shift in science process is generally following the four steps, namely: experiment, theory, computing, and data. Knowing visibility forecasting could be achieved from the physical and chemical point of view. Thus previous studies try to identify the impact of the physical and chemical mechanisms to construct a complete forecasting theory. Yet how to find representative data become a big problem the researchers are now facing, as the visibility data is too enormous. Thus it is a very imperative item to carefully treat the massive features of visibility when constructing the forecasting theories.

More understanding about the classification of big data helps us in the research of visibility. Scholars have divided the information collected into the following three categories as: (A) Structured data: such as the database. (B) Semi-structured data: such as email, blog articles. (C) Unstructured data: such as file, image, audio, video, and so on. These three forms of data will be encountered in the analysis of visibility [19, 24]. In addition, visibility can also be presented in the form of a photograph, which is the third type of information [20, 23]. No matter which part of the above types the visibility data belongs to, it is important to keep in mind the characteristics of big data contained by visibility information.

1.2 Previous Research of Visibility Forecasting, Big Data, and Computational Intelligence

Big data is a kind of massive, complex and rapidly changing information. The most important thing in the analysis of big data is not in the amount itself, but in what you can use it for. The main features of big data include the following: perception, IOT, and intelligence. So the characteristic of big data is not only represented by its massive amount, but also contains the following features as huge data and data complexity, as well as the increase of transmission speed etc. Consequently, the study of the big data is an important milestone in the forecast of visibility.

The recent progress in the field of big data includes its processing, interpretation, collection and organization. Indeed the big data processing techniques can solve many of our practical problems, as told by the past experiences. For example, WALLMART improve their marketing strategies from the big data analysis of customer and find the relationship of beer and diaper purchasing. The application of big data has emerged in many sectors such as business, industry, and not-for-profit organizations. Known from the above description, big data analysis techniques should have huge space for development applied in the visibility predictions.

How to effectively use the data is another challenge. Big data has the four major characteristics, which are: (i) massive in volume; (ii) immediacy in velocity; (iii) diversity and variety; and (iv) uncertainty or veracity. These features are included in the forecast of visibility. The four steps of data mining provide us a criterion, which are: (i) problem definition; (ii) data selection; (iii) data processing; and (iv) knowledge acquisition. Therefore, it should focus on how to extract the meaningful information from massive visibility data in future studies.

The treatment of big data is very complex, and therefore must apply the computer for the operation of this information. Conventional statistical models with computer software, has provided convenient prediction and analysis on big data. It is also essential to take advantage of the high-speed computing power to find out the relationship between each parameter within the big data. The development of artificial intelligence algorithms in recent years is also a good tool for big data analysis [10, 12, 32]. For example, the data such as customer transactions for a mega-retailer, the use of neural network analysis for environmental data and weather monitoring etc. Appropriate use of artificial intelligence algorithms can help us predict better outcomes in visibility, such as the use of neural network analysis for environmental data, and the simulation and prediction of natural disasters, etc.

1.3 Inadequacies of Previous Studies

Visibility forecasting is a problem with the consideration of time and space distribution [6]. Since there is a lot of variable affecting visibility change, the mathematical models constructed for predicting visibility should be a multiple dimensions structure. There is no such considerations in the traditional predictive model, they only consists of the sampling data for analysis from the time domain. Therefore, in predicting visibility, the use of software with the capability to handle both temporal and spatial distribution is necessary. Accordingly the characteristics of its multi-dimension must be considered in the development of future forecasting model.

The application of data mining technique is the appropriate choice for research in big data of visibility. There is a large amount of data in visibility measurement; it is possible to find many rules for forecasting within this information. The data mining has been developed for some time, so these proven technologies can provide the technique for visibility predicting. This paper will describe the current data mining techniques and discuss how to effectively use these methods. Application of data mining for big data analysis is the important item of visibility study.

1.4 Main Issues Discussed in This Research

This study explores the following questions from the perspective of big data such as: (1) current development in visibility forecasting; (2) the main difficulties and bottlenecks in visibility forecasting; (3) those prediction method applied on visibility forecasting; (4) advantages and disadvantages of these prediction methods.

Big data analysis helps us to find the optimal solution for visibility forecast. Its process is to find an optimal solution under many environmental restrictions and conflicting conditions. The most appropriate answer represents the best compromise of all the conditions. To make the computing capacity of the computer can afford, lot of meaningless information has to be sacrificed and discarded. In other words, the term "optimum" is actually a process of simplification. The traditional way of data processing is not only difficult to know the whole picture of real phenomenon, but also limits the possibility to find more real facts from existing data. The main concept of this study is shown in Fig. 1.

The main research question in this chapter is listed as follows,

- The big data analytics of environmental visibility.
- What kind of data mining methodologies can be applied in visibility forecasting?
- Application of decision trees and the optimization mechanisms for visibility forecasting.
- The computational intelligence algorithms for big data.
- The association of big visibility data and atmospheric dynamics.

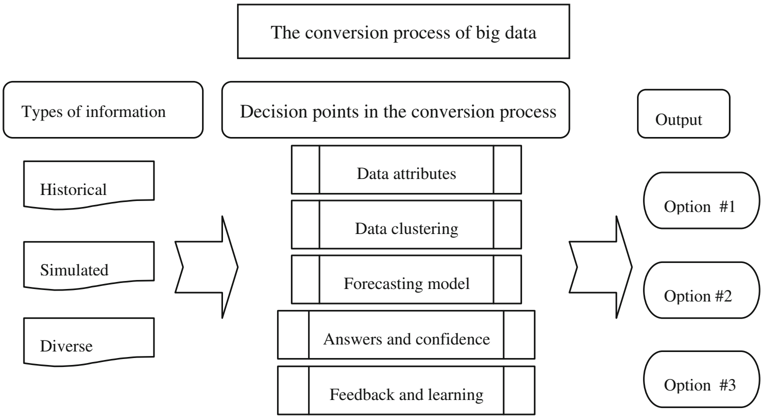


Fig. 1 Framework of big data analytic on visibility

- Case study of the applications of big data analytic in visibility forecasting.
- Future study topics and tools for big data analysis.

2 Information Granularity of the Big Visibility Data and Its Mining Technique

2.1 General Description of Data Mining Method

It is imperative to know the characteristic of visibility as a big data. The first of big data study is how to use the existing techniques and those under current development to grab the important information. The other point is as the following. How to reduce the amount of map captured from the image processing procedure? How to use the machine learning technique to find the hidden rule in the big data set? The emerging new techniques can replace the position of analysis by the traditional method. So it is worth spending time to study the application of big data analysis.

Data mining is a new analytical methods developed in recent years. It is very suitable for the application of visibility analysis, because there are many implicit rules hidden in the visibility data. These rules are helpful for us to predict the visibility change. The data mining technique help us to find out these rules. Therefore, the progress of data mining research is presented first. The most commonly used data mining methods include: classification, clustering or segmentation, association or sequence, and prediction or estimation. These methods can be appropriately used to predict the visibility.

Classification can be used to predict the visibility, because it tells us a definitive result of classification. The outcome of classification is only the circumstances of YES or NO. Classification is the job after performing the task of clustering; and it tells us which category this case belongs to. Therefore classification is the work after clustering. Classification can clearly tell us the status of the category this visibility condition belongs to.

Traditional cluster method can group or segment visibility data. Cluster method can be used to predict visibility, for example, to know what kinds of synoptic system generates the status of visibility change? Yang has already studied the class of atmospheric stability by different type and the physical principle of atmospheric dynamics. Some other scholars have focused their attention on the subject of turbulence typing. However, there is no physical principle involved in clustering or segmentation; it is directly determined by the data results displayed. It can be seen, cluster provides us a technology for clustering solely from the statistical data. This result can be mutually authenticated to the physical model.

Association or sequence is the way to find the accompanied environmental factors under certain visibility conditions, so it can be also used to predict the visibility. For example, what is the association between visibility and atmospheric stability? Is visibility related to temperature or humidity? It will be a substantial progress for the visibility forecasting if it is possible to find the mutual relation.

2.2 Commonly Used Clustering Method

The application of cluster in visibility forecasting is to find the meaningful group of visibility measurement by their similarity. Separation of these data is according to the distance between the data. The “distance” is used as a basis for classification. Those data with the closer “relative distance” and higher “degree of similarity” are classified into the same group. These statistical analysis methods do not require any physical assumptions. If it is possible to compare the results through statistical cluster analysis with the results from physical dynamic mechanism, it will be a great help for the future research of visibility forecasting [11, 43].

Applying the process of clustering analysis means we do not know the detailed classification of this group of data. Currently our understanding of visibility is just in this stage. Although there are many different parameters for analyzing and explaining, we still can directly apply the clustering techniques to analyze the visibility data. It is necessary make a detailed clustering by the characteristics of visibility through this technique. The research of visibility clustering is the focus on the study of big visibility data.

There are many methods frequently used in the cluster analysis, including: hierarchical, and non-hierarchical. The method of non-hierarchical includes: K-means, density based, and self-organizing map, SOM. The method of K-Means is to divide the data into K groups in the beginning by any individual data, and then adjust the cluster by two rules as: minimal variation within the group and maximum variation between groups. The hierarchy process includes agglomerative and divisive methods. There are two kinds of distance, the distance between the points and inter-cluster distance. Euclidean distance, Mahalanobis Distance, and city block distance are the three typical calculation methods.

2.3 Commonly Used Classification, Association, and Sequence Method

There are many methods for the classification of visibility, such as the logistic regression model, polytomors model, and ordinal logistic model. These three models can make good performance in classification for the data to be analyzed. In addition to the above method, there are other classification methods, including: discriminate analysis, classification tree, random forest, artificial neural network, SVM support vector method, and Bayesian classifier.

There are two ways for finding the cross relationship between two parameters. One is association analysis, and the other is sequence analysis. In general, the commonly used statistical software can carried out the above analysis. This is the first work to be done in forecasting the atmospheric visibility.

The quantitative relation between two parameters is the measures of association. Correlation analysis is the way to examine the linear relationship of two variables. The association between two continuous variables can be represented by the scattering plot and the coefficient of correlation. The equation to calculate the coefficient of correlation is as following.

$$r = \frac{\mathrm{cov}(x,y)}{s_x s_y} = \frac{\sum (X-\bar{X})(Y-\bar{Y})}{\sqrt{\sum (X-\bar{X})^2 (Y-\bar{Y})^2}} = \frac{SP_{xy}}{\sqrt{SS_x SS_y}}. \quad (1)$$

2.4 Commonly Used Prediction or Estimation Method

There are six methods which can be used in this section, such as: general statistical estimation and test, regression analysis, time series, logistic regression model, regression tree, and artificial neural network (ANN), and etc.

There must have a training group and forecasting group when forecasting the big data. Usually, tenth of the number were taken for prediction, and the other are used as the training data for establishing the model. The principle for selection is by random selecting so as to ensure the predicted results can be consistent with the actual situation. The accuracy of prediction results more than fifty percent is considered as the acceptable predictions.

The association analysis has been applied to the study of many environmental events. Chen and Cheng have applied the grey relational analysis to study the forecasting of typhoon [8]. Chen has also used of neural networks and fuzzy analysis to predict typhoon dynamics [4, 5]. So the association analysis can also be applied in the forecasting of atmospheric visibility. However, these outcomes should be examined by the statistical testing.

3 Forecasting of Visibility by Decision Tree and Time Series Model

This section comprises two parts, decision tree and time series. The first one tell us the definite results of visibility forecasting, and the second one give us the clear picture of the trend of future visibility variation [28].

3.1 Commonly Used Decision Tree

The application of decision tree in the study of visibility forecasting helps us to determine the weather condition and possible risk due to the circumstance. Decision tree consists of a decision diagram and the possible outcomes including resource costs and risks. It can be used to create a plan to reach the goal. A special kind of tree structure is used to assist decision-making in decision tree. A decision tree is a tree-like decision support tools graphics or decision model. It is an algorithm to display the results of random events, resources, costs and practicality [2, 9, 42].

Decision tree is often used in operations research, particularly in decision analysis; it can help to determine a strategy most likely to reach the target. According to this function, it can be divided as classification tree and regression tree. In practice, the decision often had to be adopted in the case of incomplete knowledge. A decision tree uses a parallel probabilistic model as the best choice model; the online selection model can also be used as the optional algorithm. Another use of decision tree is to calculate conditional probabilities as a descriptive tool. The form of decision tree in visibility forecasting is shown in Fig. 2.

Commonly used decision tree includes: Classification And Regression Tree (CART), CHi-squared Automatic Interaction Detection (CHAID), Quick, Unbiased, Efficient, Statistical Tree (QUEST), C4.5, Random Forest, and Cubist.

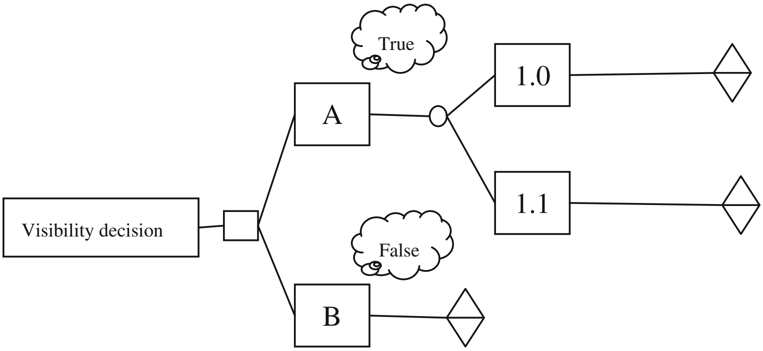


Fig. 2 A typical case of decision tree application in visibility forecasting

3.2 Commonly Used Linear and Non-linear Time Series Model

Time series is the most common method to analyze atmospheric visibility data in time domain. Because the measured visibility data is from the monitoring station, therefore we call the observational results as a "visibility time series". Visibility is a stochastic process and chaotic occurrence in the environmental system. The chaotic behavior could be investigated by many mathematical tools such as histogram, wave analysis, or spectral analysis [21, 22]. It can be linear or non-linear. The nonlinear of visibility comes from many reasons such as the scale-invariant and clustering characteristics.

Yao and Liu have used the linear and non-linear time series to predict the visibility variation in Shanghai area [33]. It is also the way to extract the hidden information from the big visibility data. The method of autoregressive integrated moving average (ARIMA) proposed by Box-Jenkins is the most frequent used time series methods [1]. The seasonal univariate ARIMA$(p, d, q)(P, D, Q)$s model is given by

$$\Phi(B)[\Delta y_t - \mu] = \Theta\,(B)\,a_t \quad \mathrm{t} = 1, \ldots, N \tag{2}$$

where

$$\Phi(B) = \varphi p(B)\Phi P(B) \tag{3}$$

$$\Theta(B) = \theta q(B)\Theta Q(B) \tag{4}$$

and μ is an optional model constant.

Independent variables x_1, x_2,…, x_m can also be included in the model. The model with independent variables is given by

$$\Phi(B)\left[\Delta\left(y_t - \sum_{i=1}^{m} c_i x_{it}\right) - \mu\right] = \Theta(B)a_t \tag{5}$$

where c_i, $i = 1, 2,\ldots, m$, are the regression coefficients for the independent variables.

Considering the variation of visibility information, so the threshold auto regression model, TAR, is considered [26, 27]. There may have many regimes among these data, so this time series is a self-exciting threshold autoregressive process which follows the equation [29–31].

$$X_t = \begin{cases} a_0^{(1)} + \sum\limits_{i=1}^{p} a_i^{(1)} X_{t-1} + \varepsilon_t^{(1)} & \text{if } \mathrm{X}_{t-d} \leqslant r_1, \\ a_0^{(2)} + \sum\limits_{i=1}^{p} a_i^{(2)} X_{t-1} + \varepsilon_t^{(2)} & \text{if } \mathrm{X}_{t-d} > r_1, \end{cases} \tag{6}$$

where t € $\{p + 1,\ldots,$ N), N is the number of observations.

For a time series with three regimes, the equation become

$$y_t = \begin{cases} \alpha_1 + \beta_1 y_{t-1} + \varepsilon_{1t} & z_{t-d} < \gamma_1 \\ \alpha_2 + \beta_2 y_{t-1} + \varepsilon_{2t} & \gamma_1 \leq z_{t-d} < \gamma_2 \\ \alpha_3 + \beta_3 y_{t-1} + \varepsilon_{3t} & z_{t-d} \leq \gamma \end{cases} \tag{7}$$

4 Forecasting of Visibility by Intelligent Computation Model

4.1 The Application of Neural Network in the Study of Visibility Forecasting

The main concept of Artificial Neural Networks, ANNs, is a computing system trying to imitate the human nervous system. It is composed by many nonlinear operations (i.e.: neuron) and the links between these arithmetic unit. This arithmetic unit generally parallel and distributed approach in computing, and use the computer hardware and software to simulate the information processing system of the biological neural network. In general, the fathering of whole ANN is in the form like the human brain. It can show the capabilities of learning, recall, induction, and deduction through the sample or trained data. The characteristics of neural network in dealing with complex work are: (1) no need for complex mathematical models to define the problem; (2) do not have to solve differential equations, integral equations, or any other mathematical equations; (3) face the complex problems and uncertainties by learning the environment [25].

There are five commonly used neural network in the study of visibility forecasting, they are Feed-Forward ANN (FFANN), Back-Propagation ANN (BPANN), Radial-Basis-Function ANN (RBFANN), Multi-Layer Perception ANN (MLPANN), and Self-organizing map ANN (SOMANN). These modes can be applied based on different requirements.

The structure of the neural network is very similar. It comprises an input layer, a hidden layer, and an output layer. The equations are as the following.

$$A_j(\bar{x}, \bar{w}) = \sum_{i=0}^{n} x_i w_{ji} \tag{8}$$

$$O_j(\bar{x}, \bar{w}) = \frac{1}{1 + e^{A(\bar{x}, \bar{w})}} \tag{9}$$

$$E_j(\bar{x}, \bar{w}, d) = \left(O_j(\bar{x}, \bar{w}) - d_j\right)^2 \tag{10}$$

$$E(\bar{x}, \bar{w}, \bar{d}) = \sum_j \left(O_j(\bar{x}, \bar{w}) - d_j\right)^2 \tag{11}$$

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} \tag{12}$$

Huang and Gu [15] has applied the Back-Propagation Neural Network on the prediction of PM_{10}. The first step of their study is variable screening. The best combination of its input is: previous day PM_{10}, humidity, wind speed, wind direction, sunshine hour, surface radiation, rainfall, atmospheric pressure and so on. The number of hidden neuron and training times were 17 and 35,000, respectively.

4.2 *The Use of Fuzzy Clustering in the Study of Visibility Forecasting*

Fuzzy clustering analysis is the result based on fuzzy theory. Zadeh [45] first proposed the fuzzy set theory, and considers it is necessary to use the theory of fuzzy concepts to deal with the problem of ambiguity and complexity. In the fuzzy set theory, the attribution of an object, is no longer the traditional dichotomy, but changed to the expression of membership function. The membership degree is a collection of vague impression of objects. So the feature the original has a discrete range {0, 1} function can be extended into the continuous range [0, 1] of the membership function. The value determined by membership functions are namely membership degree.

This method can help us to analyze the visibility data when combined with computer with the high-speed processing power for big data. Applying the “membership grade” to replace the previous “binary logic” approach inference process can keep the original message of data. The original message will not be lost resulting in the inference process.

This fuzzy clustering method can be in line with the time-series forecasting models, so there has been lot of researches applied to environmental issues. For example, Huang and Chen [16] have used this model to conduct a study of ground water quality statistics in Taiwan. Chang and Chang [3] used this method to judge marine pollution of Kaohsiung. Chen, based on the method of fuzzy time series, conducted a study to predict the suspended particulate concentration, which is a precursor of atmospheric visibility.

Visibility time series can be predicted by the fuzzy theory, as shown in the following equation.

$$\mathrm{X_{vis}} = \{\mathrm{x_{vis}t},\ \mathrm{t}\ = 1, \ldots, \mathrm{N}\ \} \tag{13}$$

A fuzzy time series of visibility can be estimated by the event characterization function, ECF, as

$$\mathrm{ECF} = \mathrm{g(t)} =\ \mathrm{g}\ (\mathrm{X_i}, \mathrm{X_{i-1}}, \ldots \mathrm{X_l}) \tag{14}$$

The ECF can be chosen as:

$$g(t) = \frac{X_{i+1} - X_i}{X_i} \tag{15}$$

If visibility prediction is needed to do one step before, the event characterization functions in Eq. (15) can be selected. The membership function of the fuzzy time series, *f*(*x, a, c*), was given by the following equation where *a,* and *c* are the mapping on a vector x.

$$f(x, a, c) = \frac{1}{1 + e^{-a(x-c)}} \tag{16}$$

The value in time step (n + 1) is determined by the membership function *f*(*x*, *a*, *c*), and the value of previous step as follows,

$$\mathrm{X_{vis}}(\mathrm{t_{n+1}}) =\ \mathrm{X_{vis}}(\mathrm{t_n}) + f(x, a, c) \cdot [\ \mathrm{X_{vis}}(\mathrm{t_{n+1}}) -\ \mathrm{X_{vis}}(\mathrm{t_n})\,] \tag{17}$$

5 Case Study and Tool for Big Data Visibility Forecasting

5.1 Case Study of the Big Data Visibility Forecasting by Time Series Model

A study to forecast atmospheric visibility using the whole year hourly data was presented, it is a huge amount for the whole year's hourly data. Five experiments were performed in this study using time series model to predict the big data of visibility. The first was 1-step ahead hourly model (1SH). The second was 1-step ahead daily model (1-SD). The third was 24-step ahead hourly method (24-SH). The fourth was the 24-step ahead hourly method (24-SH). And finally the fifth was the threshold autoregression model (TAR).

Model A: 1-step ahead hourly model (1SH)
This model forecast the next hourly average value on the basis of the established forecasting derived from the previous data warehouse. When forecasting the value of the next hour, the actual observed value replace the preceding forecast value.

Model B: 1-step ahead daily model (1-SD)
This model is based on the previous daily average value of observation. The next daily averaged value was predicted by forecasting model derived from daily averaged value. The same procedure was applied as Model A when forecasting the next day value.

Model C: 24-step ahead hourly model (24-SH)
This model forecasts value in the next 24 h by repeating the procedure in Model A without repeating the real value in each value in each 1-step ahead hourly mo1-SH forecasting.

Model D: 7-step ahead daily model (7-SD)
This model forecasts value in the next 7 day based on the previous daily average value. The procedure is the same as Model C except the target of forecasting changed into the next 7 day.

Model E: Threshold autoregression model (TAR)
This model forecasts the non-linear characteristic of time series by assuming there are regimes and estimating the threshold for dividing the regimes of the time series (Tables 1 and 2).

5.2 Case Study of Visibility Forecasting by Neural Network Study

Three cases of neural network forecasting to predict the visibility are performed in this study. The first is the backward propagation artificial neural network (BPANN), the second is genetic algorithm artificial neural network (GAANN), and the third is fuzzy artificial neural network (FANN). The results with the observed data for one month are shown in Fig. 3.

5.3 Tool for Big Data and Intelligent Computing of Visibility

There are a lot of commercial softwares, such as SPSS, SAS, Eview and etc. However, it will be convenient for the beginner or students to find the available

Table 1 The average value of visibility forecasting

Year/ Month	$E(Y_t) = \mu_y$ average	Weighting factor	$V(Y_t) = \sigma_y^2$	σ_y = Standard deviation	Skewness	Kurtosis
December	5,784	0.567615	22381242	3,869	1.147867	0.935077
January	6,378	0.625908	22442698	3,518	0.72337	0.401553
February	7,408	0.726987	31907349	4,306	0.552326	−0.35657
March	8,108	0.795682	35027158	4,319	0.421546	−0.6406
April	9,138	0.896762	45820650	5,013	0.332707	−0.94555
May	11,316	1.1105	64317144	5,683	0.009726	−1.28267
June	8,477	0.831894	47986537	5,660	0.871286	−0.52456
July	12,972	1.273 > 013	83054233	6,402	−0.21163	−1.49991
August	15,546	1.525613	87534118	5,630	−0.82444	−0.84129
September	15,356	1.506968	85856502	5,590	−0.85271	−0.67108
October	11,976	1.17527	65434036	5,463	−0.0963	−1.24378
November	10,062	0.987439	46835391	4,654	0.23217	−0.81876
Whole year	10190	1	60845761	5,970	0.353471	−1.1582

Table 2 Tabulation of the statistical properties for the four models

Model type		Model A: (1SH)	Model B: (1-SD)	Model C: (24-SH)	Model D: (7-SD)
Model description		1-step ahead hourly model	1-step ahead daily model	24-step ahead hourly method	7-step ahead daily model
n	Sample	8,439	361	364	52
μ_y	Mean	10,190	10,101	9,383	10,102
M_d	Median	9,060	9,586	8,595	9,279
Max	Maximum	20,000	20,000	20,000	18,682
Min	Minimum	80	1,550	180	2,158
σ_y	Standard	deviation	5,971	4,811	5,564
	4,392				
S_k	Skewness	0.356328	0.353470	0.509404	0.259623
K_t	Kurtosis	1.841814	2.221009	2.791018	2.069006
J-B	Prob.	Jarque-Bera probability	650.2502	16.64497	25.52177
	2.462172				

software for their practices. From this point of view, R can be a good resource for application. R is a free software environment which is very convenient for statistical computing and graphics. It can be obtained for free download online. It compiles and runs on a wide variety of platforms, such as UNIX, Windows and MacOS. R is a freeware code on-line and Table 3 is the suggested software in R for the big data analysis.

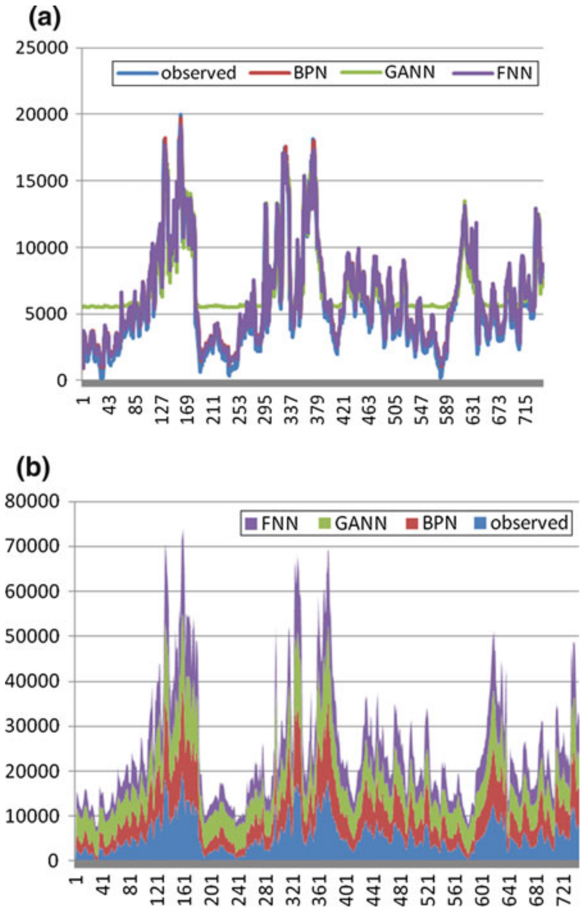


Fig. 3 Big data forecasting results comparison of visibility from three intelligent computing method: the backward propagation artificial neural network BPANN, the genetic algorithm artificial neural network (GAANN), and the fuzzy artificial neural network (FANN)

6 Conclusions

Scientists accustomed to the way of data processing based on "hindsight analysis for prior forecast", which means, to analyze the "known unknowns". The analysis of big data provide us the source of "unknown unknowns" In other words, because the researchers can quickly make dozens of patterns in visibility changes from the raw data, it allows the data users began to have the additional option of "data". The researchers want appropriate inputs to provide different forms of output. In the present study, it has been done trying to predict the visibility by the above analysis.

This Chapter is aimed at the intelligent computation of big data to the forecasting of environmental visibility according to the given the theme of this book. In this study, the conceptual description of big data applied in the prediction of visibility is presented. With examples and applications, thereby this study expands our reflection and imagination on big data, it also stimulates the imagination and

Table 3 Software in R for the intelligent computing of big data

Type		No.	Library
A	Cluster	1	mlbench
		2	cluster
		3	SOMbremo
		4	birch
B	Association	1	arules
		2	IsoGene
		3	biglars
			biglm
C	Decision tree	1	tree
		2	CHAID
		3	Cubist
		4	**r**andom Forest,
		5	mlbench
		6	CART
		7	QUEST
		8	Mob, Cubist.
D	Artificial neural network	1	nnet
		2	neuralnet
		3	RSNNS
			PopGenome
E	Big data	1	bigmemory
		2	biganalytic
		3	bigtable
		4	bigGP
		5	bigrf

explicability of knowledge in this field. How to realize the full potential of big data and computation intelligence in the forecasting of visibility, as well as the power of information and network technology become more imperative. Obtaining more visibility information is not a problem now, however, treatment of these big data help us to re-think the environmental phenomenon and the meaning of these information.

References

1. Box, G.E.P., and Jenkins, G.: Times Series Analysis, Forecasting and Control. Holden-Day, San Francisco (1970)
2. Cha, S.H., Tappert, C.C.: A genetic algorithm for constructing compact binary decision trees. J. Pattern Recogn. Res. **4**(1), 1–13 (2009)
3. Chang, D.C., Chang, Y.C.: Application of fuzzy cluster algorithms in pollutant's spatial distribution of Kaohsiung coast. Master thesis, National Sun-Yat Sen University, Kaohsiung, Taiwan (2002)

4. Chen, W.K.: Study of the pattern recognition of particulate concentration by fuzzy time series and neural network analysis. Paper presented at the 2009 cross strait conference on aerosol science and technology (2009)
5. Chen, W.K.: An approach to pattern recognition by fuzzy category and neural network simulation. Paper presented at the 2010 international conference on machine learning and cybernetics (ICMLC) (2010)
6. Chen, W.K.: Environmental applications of granular computing and intelligent systems granular computing and intelligent systems. Intel. Syst. Ref. Libr. **13**(2011), 275–301 (2010)
7. Chen, W.K., Wang, P.: Numerical modeling of gas-phase kinetics in formation of secondary aerosol. China Particuology **5**(4), 267–273 (2007). doi:10.1016/j.cpart.2007.05.002
8. Chen, W.K., Cheng, C.S.: Application of grey theory in predicting the disaster loss of typhoon. In: 2012 global Chinese conference on environment and energy, Hsinchu, Taiwan 2012
9. Deng, H., Runger, G., Tuv, E.: Bias of importance measures for multi-valued attributes and solutions. In: Proceedings of the 21st international conference on artificial neural networks (ICANN) (2011)
10. Dubois, D., Prade, H.: Fuzzy Sets and Systems. Academic Press, New York (1988)
11. Gil, J.M., Gracia, A., Sanchez, M.: Market segmentation and willingness to pay for organic products in Spain. Int. Food Agribusiness Manage. Rev. **3**(2), 207–226 (2000)
12. Goguen, J.A.: L-fuzzy sets. J. Math. Anal. Appl. **18**, 145–174 (1967)
13. Hsu, N.C., Herman, J.R., Bhartia, P.K., Seftor, C.J., Torres, O., Thompson, A.M., Holben, B.N.: Detection of biomass burning smoke from TOMS measurements. Geophys. Res. Lett. **23**(7), 745–748 (1996). doi:10.1029/96gl00455
14. Hsu, N.C., Herman, J.R., Torres, O., Holben, B.N., Tanre, D., Eck, T.F., Lavenu, F.: Comparisons of the TOMS aerosol index with Sun-photometer aerosol optical thickness: results and applications. J. Geophys. Res. Atmos. **104**(6), 6269–6279 (1999). doi:10.1029/1998jd200086
15. Huang, L.M., Gu, W.J.: Application of back-propagation neural network on the prediction of PM_{10}. J. Soil Water Conserv. **44**(4), 341–360 (2012)
16. Huang, L.C., Chen Y.Y.: Statistical study of the ground water quality in Taiwan. Master thesis, National Central University, Chungli, Taiwan 2000
17. Lee, C.G., Yuan, C.S., Chang, J.C., Yuan, C.: Effects of aerosol species on atmospheric visibility in Kaohsiung City, Taiwan. J. A&WMA **55**, 1031–1041 (2005)
18. Lee, C.G., Yuan, C.S.: Visibility, synoptic meteorology and air pollutants in Kaohsiung City, Taiwan. In: Presented at the 5th Asian Aerosol Conference, Kaohsiung, 26–29 August 2007
19. Luo, C.H., Liu, S.H., Chen, Y.S., Yuan, C.S.: Measure atmospheric visibility by the digital telephotography. In: Presented at Seventh International Conference on Atmospheric Sciences and Applications to Air Quality and Exhibition, Taipei 2000
20. Luo, C.H., Liu, S.H., Yuan, C.S.: Measuring atmospheric visibility by digital image processing. Aerosol Air Qual. Res. **2**(1), 23–29 (2002)
21. Luo, C.H., Yuan, C.S., Wen, C.Y., Liaw, J.J., Chiu, S.H.: Investigation of urban atmospheric visibility using Haar wavelet transform. Aerosol Air Qual. Res. **5**(1), 39–47 (2005)
22. Luo, C.H., Wen, C.Y., Yuan, C.S., Liaw, J.J., Lo, C.C., Chiu, S.H.: Investigation of urban atmospheric visibility by high-frequency extraction: model development and field test. Atmos. Environ. **39**, 2545–2552 (2005)
23. Luo, C.H., Lin, K.H., Wen, C.Y., Chiu, S.H., Yuan, C.S.: Suburban atmospheric visibility sensing by an image degradation processor: model development and field test. Int. J. Remote Sens. **32**(24), 9801–98102 (2011)
24. Luo, C.H., Yuan, C.S.: Investigation on Taipei atmospheric visibility by an image processor. In: Presented at the 5th Asian Aerosol Conference, Kaohsiung, 26–29 August 2007
25. Pedrycz, W.: Granular computing: analysis and design of intelligent systems. CRC Press/Francis Taylor, Boca Raton (2013)
26. Tsay, R.S.: Testing and modeling threshold autoregressive processes. J. Am. Stat. Assoc. **84**(405), 231–240 (2006)

27. Tsay, R.S.: Nonlinearity tests for time series, Biometrika **73**, 461–466 (1988). doi: 10.1093/biomet/73.2.461
28. Tsay, R.S.: Outliers, level shifts, and variance changes in time series. J. Forecast. **7**, 1–20 (1988)
29. Tong, H.: On a threshold model in pattern recognition and signal processing. In: Chen, C.H. (ed.), Sijhoff & Noord-hoff, Amsterdam (1978)
30. Tong, H.: Threshold models in nonlinear time series analysis (lecture notes in statistics no. 21). Springer, New York (1983)
31. Tong, H., Lim, K.S.: Threshold autoregression, limit cycles and cyclical data (with discussion). J. R. Stat. Soc. Ser. B **42**, 245–292 (1980)
32. Wang, H.O., Tanaka, K., Friffin, M.F.: An approach to fuzzy control of nonlinear systems stability and design issues. IEEE Trans. Fuzzy Syst. **85**, 305–309 (1996)
33. Yao, J., Liu, W.: Nonlinear time series prediction of atmospheric visibility in Shanghai. In: Pedrycz, W., Chen, S.-M. (eds.), Time Series Analysis, Modeling and Applications, vol. 47, pp. 385–399, Springer, Berlin (2013)
34. Yang, H.Y., Yuan, C.S.: The correlation of the visibility variation with weather patterns and meteorological factors in the south of Taiwan. 96th AW&MA annual meeting, San Dieago, California 2003
35. Yuan, C.S., Lee, C.G., Chang J.C., Liu, S.H. Yuan, C. Yang, H.Y.: Correlation of atmospheric visibility with chemical composition and size distribution of aerosol particles in urban area. 93rd Air and Waste Management Association annual meeting, Salt Lake City, Utah 2000
36. Yuan, C.S., Lee, C.G., Liu, S.H., Chang, F.T.: Innovative measurement of visual air quality and its correlation with meteorological factors and air pollutants in subtropics. In: Presented at Seventh International Conference on Atmospheric Sciences and Applications to Air Quality and Exhibition, Taipei 2000
37. Yuan, C.S., Lee, C.G., Liu, S.H., Chang, F.T.: Characterization of urban air quality using atmospheric visibility as an indicator: feasibility study. In: Proceedings of 2000 International Conference on Air Quality Management, pp. 102–127, Kaohsiung 2000
38. Yuan, C.S., Lee, C.G., Liu, S.H., Chang, J., Yuan, C., Yang, H.Y.: Correlation of atmospheric visibility with chemical composition of Kaohsiung aerosols. Atmos. Res. **82**, 663–679 (2006)
39. Yuan, C.S., Wong, C.H., Yuan, C., Lee, C.G., Lee, Y.C.: Feasibility Study of the applications of alternative and supplementary air quality index in Kaohsiung City. In: Proceedings of the Seminar on commissioned projects of Environmental Protection Bureau, Kaohsiung City Government in 2001, pp. 4-1–4-29, Kaohsiung 2001
40. Yuan, C.S., Huang, M.H., Lin, Y.C., Cheng, S.W.: Influence of dust storm from Mainland China to the characteristic of suspended particulate in Penghu area. In: 9th International Conference of Aerosol Science and Technology, pp. 14–18, Yunlin 2001
41. Yuan, C.S., Lin, Y.C., Lee, C.G.: the application of artificial neural network technology for the prediction of ambient particulate matter concentration. In: 96th Air and Waste Management Association Annual Meeting, San Diego 2003
42. Yuan, Y., Shaw, M.J.: Induction of fuzzy decision trees. Fuzzy Sets Syst., 125–139 (1995)
43. Yuksel, A., Yuksel, F.: Measurement of tourist satisfaction with restaurant services: a segment-based approach. J. Vacation Mark. **9**(1), 52–68 (2002)
44. Yuan, C.S., Lo, C.C., Chen, W.C., Chen, C.C., Luo, C.H., Yuan, C., Yang, H.Y.: Forecasting atmospheric visibility of metropolitan area using an automated observation system. In: Presented at the 97th Air and Waste Management Association annual meeting, Indianapolis, Indiana 2004
45. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**(3), 338–353 (1965)

Application of Computational Intelligence on Analysis of Air Quality Monitoring Big Data

Tzu-Yi Pai, Moo-Been Chang and Shyh-Wei Chen

Abstract For controlling air pollution, the Taiwan Environmental Protection Administration (TEPA) installed automatic air quality monitoring stations (AQMSs) and TEPA prescribed the industries to install continuous emission monitoring systems (CEMS). By 2014, there were a total of 76 AQMS and 351 CEMS in the entire nation. Therefore, the huge amount of air quality monitoring data forms big data. The processing, interpretation, collection and organization of air quality monitoring big data (AQMBD) have emerged in air quality control including industry management, traffic reduction, and residential health. In this chapter, the application of computational intelligence on analysis of air quality monitoring big data was reviewed worldwide. Additionally, the application of computational intelligence (CI) including artificial neural network, fuzzy theory, and adaptive network-based fuzzy inference system (ANFIS) was discussed. Finally, the implementation of CI on AQMBD granular computing was proposed.

Keywords Computational intelligence · Air quality monitoring big data · Artificial neural network · Swarm intelligence

T.-Y. Pai (✉)
Master Program of Environmental Education and Management,
Department of Science Application and Dissemination, National Taichung
University of Education, Taichung 40306, Taiwan, Republic of China
e-mail: bai@ms6.hinet.net

M.-B. Chang
Institute of Environmental Engineering, National Central University,
Chungli 32001, Taiwan, Republic of China

S.-W. Chen
Environmental Protection Bureau, Taoyuan County Government,
Taoyuan 33001, Taiwan, Republic of China

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_21,

1 Introduction

After the commencement of the usage for Internet in the 1970s and the World Wide Web in the 1990s, various data generation and collection speeds have increased exponentially. Recently, the big data era has crept into many fields, from governments, industries, and health organizations to environment science.

In the past two decades, air pollution has improved in most cities in Western Europe, North American as well as Japan [4]. Although improvements are also achieved in Taiwan, the regulation efficiencies of stationary pollution sources and mobile pollution sources are still not significant because their emitted characteristics [19]. The emission and ambient concentration of those regulated pollutants are potentially harmful to the health or well-being of human, animal or plant life, or to ecological systems [10, 13, 23].

For examples, carbon monoxide (CO) can cause chronic poisoning which reveals its first symptoms as headaches, blurry vision, difficulty in concentration, and confusion [1]. The results of recent epidemiologic studies also indicated that ground level ozone (O_3) can exacerbate asthma symptoms even at concentrations lower than 80 ppb (8-h average) [9]. The evidences from various studies have shown that micron particle is associated with morbidity and mortality rates particularly due to cardiovascular and respiratory illness [8].

In order to control air pollution, the Air Pollution Control Act (APCA) of Taiwan was signed in 1975. In APCA, Article 13 of Chapter "Information Mining for Big Information" prescribes that competent authorities shall select the appropriate locations for installation of air quality monitoring stations (AQMS). Article 22 of Chapter "Information Granules Problem: An Effective Solution of Real-Time Fuzzy Regression Analysis" prescribes that those public and private premises possessing stationary pollution sources shall install continuous emission monitoring systems (CEMS) to continuously monitor their operations or air pollutant emissions conditions, and shall connect their CEMS via the Internet to the competent authority.

By 2014, there were a total of 76 AQMS and 351 CEMS in the entire nation. For AQMS, the hourly air pollutant concentrations at least included 16 pollutants. The meteorological conditions included at least 4 items. For CEMS, the air pollutant concentrations per 15 min at least included 16 basic pollutants, several volatile organic compounds (VOCs), and operation parameters.

Furthermore, the data from AQMS and CEMS can be compared with the data from the National Health Insurance Research Database (NHIRD) (regulated by Ministry of Health and Welfare), real-time Electronic Toll Collection (ETC) system, Meteorological Monitoring Network (MMN) system (regulated by Ministry of Transportation and Communications), Agricultural Long-term Ecological Research Wireless Sensing Network (ALERWSN, regulated by Council of Agriculture), or National Park Environment Monitoring (NPEM, regulated by Ministry of Interior). Therefore, the huge amount of data flow forms air quality monitoring big data (AQMBD). With a large amount of mobile, web-based, and sensor-generated data approaching at a terabyte scale, the processing, interpretation, collection and organization of AQMBD have emerged in air quality control including industry management, traffic reduction, and residential health.

For solving AQMBD, the granular computing is a suitable method. Through a series of learning and training process, the overwhelming amount of data can be reorganized by different granularities, the complexity and difficulty of the calculation can be reduced, and the accuracy of classification can be improved significantly. Furthermore, granular computing can be used to deal with the classification problems with incomplete information system. Because of mimicking the human cognitive behaviors, the granular computing methods can also improve the quality of classification algorithm effectively [2, 6, 27–30, 39–42].

In this chapter, the formation and characteristics of AQMBD will be investigated. The granular computing will be reviewed. The application of computational intelligence (CI) on analysis of AQMBD will be reviewed worldwide. Additionally, the application of CI including artificial neural network (ANN), fuzzy logic (FL), and adaptive network-based fuzzy inference system (ANFIS) will be discussed. Finally the implementation of CI on AQMBD granular computing will be proposed. Actually, no study on AQMBD has been conducted in relation to granular computing. This study represents the first report to suggest the granular computing for implementation of CI on AQMBD.

2 Formation of Air Quality Monitoring Big Data

2.1 Regulation and Control Measurements

In APCA, Article 13 of Chapter "Information Mining for Big Information" prescribes that the central competent authority should establish AQMSs and should regularly publicly report the state of air quality in cities, towns, and townships where petrochemical industrial areas are located and at appropriate points selected by competent authorities at all levels. In addition, Article 22 of Chapter "Information Granules Problem: An Effective Solution of Real-Time Fuzzy Regression Analysis" prescribes that those public and private premises possessing stationary pollution sources designated and officially announced by the central competent authority shall complete the installation of automatic monitoring facilities by the designated deadline in order to continuously monitor their operations or air pollutant emissions conditions, and shall apply to the competent authority for authorization; those that have been designated and officially announced as being required to connect via the Internet shall complete the connection of their monitoring facilities via the Internet to the competent authority by the designated deadline.

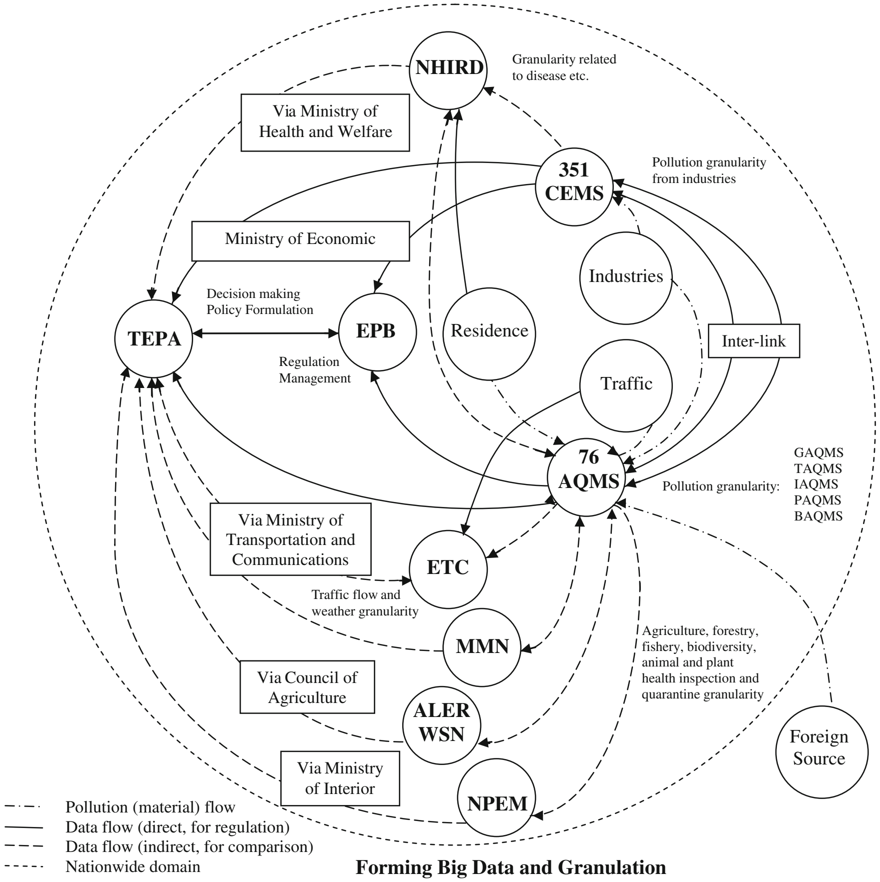

NHIRD
Granularity related to disease etc.
Via Ministry of Health and Welfare
351 CEMS
Pollution granularity from industries
Ministry of Economic
Industries
Decision making
Policy Formulation
TEPA
EPB
Residence
Inter-link
Regulation
Management
Traffic
76 AQMS
Pollution granularity:
GAQMS
TAQMS
IAQMS
PAQMS
BAQMS
Via Ministry of Transportation and Communications
ETC
Traffic flow and weather granularity
MMN
Agriculture, forestry, fishery, biodiversity, animal and plant health inspection and quarantine granularity
Via Council of Agriculture
ALER WSN
Foreign Source
Via Ministry of Interior
NPEM
Pollution (material) flow
Data flow (direct, for regulation)
Data flow (indirect, for comparison)
Nationwide domain
Forming Big Data and Granulation

Fig. 1 Granulation of AQMBD flow in nation domain of Taiwan

From 1980, the Environmental Protection Administration in the Central Government of Taiwan (TEPA) began installing automatic AQMSs and by 2014, there were a total of 76 AQMSs and 351 CEMSs in the entire nation. Furthermore, the data from AQMS and CEMS can be compared with the data from the NHIRD or real-time ETC system. Therefore, the huge amount of air quality monitoring data forms big data, as shown in Fig. 1.

2.2 Monitoring Stations and Monitoring Items

According to the Article 11 in Air Pollution Control Act Enforcement Rules (APCAER), AQMS designated in Article 13 of APCA shall include five types of AQMS including general AQMS (GAQMS), traffic area AQMS (TAQMS),

Table 1 The established locations and purposes of different AQMSs

AQMS	Established locations and purposes
GAQMS	Established in areas that are densely populated or that may be subject to high pollution or that are able to reflect the air quality distribution in a larger region
TAQMS	Established in areas of heavy traffic
IAQMS	Established in windy downwind areas industrial parks
PAQMS	Established at appropriate sites in national parks
BAQMS	Established in areas where there is relatively little human pollution or in windy downwind areas in total quantity control zones

industrial area AQMS (IAQMS), national park AQMS (PAQMS), and background AQMS (BAQMS). The established locations and purposes of different AQMSs are listed in Table 1.

The number of AQMSs established shall conform to the following principles: (1) In accordance with population and habitable area (buildings, paddies, upland fields), one general AQMS shall be established for every 300,000 persons in areas with a population density exceeding 15,000 persons per square kilometer; one general AQMS shall be established for every 350,000 persons in areas with a population density below 15,000 persons per square kilometer. The number of AQMSs may be increased in special municipalities. (2) The number of other types of AQMSs shall depend on actual needs.

The central competent authority may establish a monitoring center connected with monitoring stations in light of actual needs. The establishment of AQMS sampling orifices shall conform to the following principles. First, sampling orifices may not be in locations directly affected by pollution from flues or exhaust outlets, etc. Second, avoid disturbance of air flow and pollutant concentration by nearby obstacles. Third, avoid nearby buildings or obstructing surfaces that may affect pollutant concentration. Forth, determine the height of the sampling orifice above the ground in accordance with the vertical concentration distribution of pollutants near the monitoring station.

Article 13 of APCAER prescribes the pollutant items which shall be tested in different AQMSs. The hourly air pollutant concentrations at least included methane (CH_4), carbon monoxide (CO), carbon dioxide (CO_2), non-methane hydrocarbon (NMHC), nitrogen monoxide (NO), nitrogen dioxide (NO_2), nitrogen oxides (NOx), fine particulates ($PM_{2.5}$ and PM_{10}), and sulfur dioxide (SO_2). The meteorological conditions included ambient temperature (Temp), rainfall, relative humidity (RH), wind speed (WS), and ultraviolet B (UVB). The analytical methods for SO_2, CO, O_3, PM_{10}, NO_2 and NMHC are ultraviolet fluorescence method, nondispersive infrared method, ultraviolet absorption method, β-ray attenuation method (or tapered element oscillating microbalance technology), chemiluminescence method and flame ionization detector, respectively.

2.3 Characteristics of AQMBD

Although AQMS and CEMS often report the hourly and quarterly (15 min) average values of different air quality index, the automatic detectors equipped in AQMS and CEMS usually record the value per second. Therefore, the amount of AQMBD is very huge. Because the huge amount of data generated from automatic detectors must be transferred to the interfaces of both AQMS and CEMS, subsequently to local environmental protection bureau (EPB) and TEPA in a very short time, the speed of data in and out is very high. Besides, the data come from residences, industries, and traffic area, the data also fall into many categories of physical properties, chemical compounds, or meteorological conditions, so the range of data types and sources is very broad. In summary, the characteristics of AQMBD fit the properties of big data including volume, velocity, and variety (3Vs).

3 Application of Computational Intelligence

Since AQMBD is high volume, high velocity, and/or high variety information assets, they require new forms of processing to enable enhanced decision making, insight discovery and process optimization. In the field of CI, granular computing can simplify complex problems by granulation of overwhelming amount of data. The basic methods include computing with word, computing with fuzzy sets, computing with rough sets, and computing with quotient space. New methods derived from basic methods include uniform granular computing, efficient granular computing, novel granular computing, dynamic granular computing, and data-driven granular computing etc. [2, 6, 27–30, 39–42]. Since there is no application of granular computing on AQMBD at present, the application of CI on AQMBD is described as follows.

CI provides solutions for complicated problems. It primarily contains ANN, FL, and evolutionary computation (EC). Additionally, CI also includes biomimicry algorithms such as swarm intelligence and artificial immune systems, which can be regarded as a part of EC. Furthermore other forms including chaos theory, Dempster–Shafer theory, grey system theory (GST) and many-valued logic are used in the establishment of computational models. CI has been successfully used to predict environmental index, for instance ANNs to predict water quality, environmental materials, and air pollutant levels [19–26]. Some present applications are described as follows.

3.1 Prediction

As mentioned previously, the air pollution addresses much attention in modern society. Prediction of pollutant levels is an important and popular research topic in atmospheric environment science today. CI can be applied to several categories of prediction problems, such as particulate matters, nitrogen oxides, and photochemical oxidants etc.

Among the CI algorithms, ANN is the most popular one. The ANN modeling approach in which the important operation features of human nervous system is simulated attempts to solve problems by using information gained from past experience to new problems. In order to operate analogous to a human brain, many simple computational elements called artificial neurons that are connected by variable weights are used in the ANN. With the hierarchical structure of a network of interconnected neurons, an ANN is capable of performing complex computations, although each neuron, alone, can only perform simple work. The multi-layer perceptron structure is commonly used for prediction among the many different types of structures.

A typical neural network model consists of three independent layers: input, hidden, and output layers. Each layer is comprised of several operating neurons. Input neurons receive the values of input parameters that are fed to the network and store the scaled input values, while the calculated results in output layer are assigned by the output neurons. The hidden layer performs an interface to fully interconnect input and output layers. The pattern of hidden layer to be applied in the hierarchical network can be either multiple layers or a single layer. Each neuron is connected to every neuron in adjacent layers before being introduced as input to the neuron in the next layer by a connection weight, which determines the strength of the relationship between two connected neurons. Each neuron sums all of the inputs that it receives and the sum is converted to an output value based on a predefined activation, or transfer, function. For prediction problems, a supervised learning algorithm is often adopted for training the network how to relate input data to output data. In recent years, the backpropagation algorithm is widely used for teaching multi-layer neural networks. Traditionally, the algorithm uses a gradient search technique (the steepest gradient descent method) to minimize a function equal to the mean square difference between the desired and the actual network outputs [3, 5, 7, 11, 12, 14–18, 26, 31–38]. These studies have shown that the ANN approach is effective in simulating the dynamics of unsteady time series due to its special non-assumable, non-parametric, noise-tolerant and high-adaptive properties. ANN models can be utilized to map any non-linear function without prior assumptions on the raw data. There are several types of ANN including backpropagation neural network (BNN), radial basis function neural network (RBFNN), modular neural networks (MNN), competitive neural network (CNN), Kohonen self-organizing map neural network (KSOMNN), and Hopfield neural network (HNN) etc.

Wang et al. [36] combined the adaptive radial basis function neural network (ARBFNN) with statistical characteristics of ozone in Hong Kong specific areas, and forecast the daily maximum ozone concentration level. The improved method is trained and testified by hourly time series data collected at three AQMSs during 1999 and 2000. The results showed the effectiveness and the reliability of the proposed method.

Gautam et al. [12] developed a new technique to predict the chaotic time series of O_3 based on the ANN methods. They found that the mean absolute percentage errors

(MAPEs) lay between 12.26 and 24.01 % using ANN and 9.46–13.55 % using new algorithm. In the study proposed by Cai et al. [5], ANN was used to predict hourly air pollutant concentrations beside the city main street. The results indicated that the MAPE for predicting O_3 fell in the range of 32.93 and 45.15 %, RMSE were between 9.5 and 10.3, and R lay between 0.941 and 0.951, respectively.

Lu et al. [18] utilized the neural network of multi-layer perceptron (MLP) to predict the tendency and the effect of the pollutants including respirable suspended particulate (RSP) and nitrogen oxides (NO_2 and NO_x) from heavy vehicles and massive transportation due to domestic diesel fuel usage. Lu et al. [18] developed an improved ANN in which the principal component analysis (PCA) technique and the RBFNN were combined, and predicted the pollution in accordance with the recorded data. In this study, PCA was used to reduce and orthogonalize the raw input parameters, and these treated parameters were subsequently regarded as new input parameters in RBFNN. This proposed model was evaluated using hourly time series of RSP, NO_2, and NO_x concentrations investigated at a traffic AQMS in Hong Kong during the year 2000. Comparing with the general ANN, the network architecture, training speed, and predicting performance of the PCA RBFNN was simpler, faster and more satisfactory, respectively. By comparing the actual RSP, NO_2, and NO_x concentrations recorded at the AQMS with the predicted values of these pollutants, the prediction performance of PCA RBFNN was proven. Therefore, Lu et al. [18] concluded that the model was an effective method to predict air quality and had advantages over the general ANN.

Wang et al. [37] adopted the adaptive RBFNN (ARBFNN) and improved support vector machine (SVM) in atmospheric sciences. They employed PCA technique to the ARBFNN to fasten the learning procedure. Comparing the proposed model with the general ANN, the model could automatically determine the size of network and parameters, fasten the learning process, and achieve good prediction performances.

Although ANN can predict the target pollutants successfully, traditional neural network schemes still have several limitations which result from the possibility of getting trapped in local minimum, and the choice of model architecture. If the predicting performance can be further promoted, a better operation strategy can be formed. To overcome these limitations of traditional ANNs, and to increase their reliability, many new training algorithms have been proposed such as ANFIS. ANFIS's architecture consists of both ANN and FL including linguistic express of membership functions (MFs) and if-then rules.

Both ANN and fuzzy logic are adopted in ANFIS's architecture in which if-then rules with appropriate MFs and the specified input–output pairs are used. The learning algorithms of neural network are used for ANFIS training. Two methods are employed for updating MF parameters in ANFIS learning: (1) backpropagation for all parameters (steepest descent method), and (2) backpropagation for the parameters associated with the input MFs and least squares estimation for the parameters associated with the output MFs. Subsequently, the training errors decrease, at least locally, during the learning procedure. The more the initial MFs resemble the optimal ones, the more quickly the training parameters converge.

The fuzzy inference system with three inputs (I_1, I_2 and I_3) and one output (O_f) is taken for example to explain the ANFIS architecture in this study. Considering a first order Sugeno type of fuzzy model, the if-then rule base can be expressed as

$$\begin{aligned}
\text{Rule 1}: &\text{If } \mathrm{I}_1 \text{ is } \mathrm{A}_1 \text{ and } \mathrm{I}_2 \text{ is } \mathrm{B}_1 \text{ and } \mathrm{I}_3 \text{ is } \mathrm{C}_1,\\
&\text{Then } f_{1,1,1} = \alpha_{1,1,1}\cdot \mathrm{I}_1 + \beta_{1,1,1}\cdot \mathrm{I}_2 + \gamma_{1,1,1}\cdot \mathrm{I}_3 + \eta_{1,1,1}\\
\text{Rule 2}: &\text{If } \mathrm{I}_1 \text{ is } \mathrm{A}_1 \text{ and } \mathrm{I}_2 \text{ is } \mathrm{B}_1 \text{ and } \mathrm{I}_3 \text{ is } \mathrm{C}_2,\\
&\text{Then } f_{1,1,2} = \alpha_{1,1,2}\cdot \mathrm{I}_1 + \beta_{1,1,2}\cdot \mathrm{I}_2 + \gamma_{1,1,2}\cdot \mathrm{I}_3 + \eta_{1,1,2}\\
\text{Rule 3}: &\text{If } \mathrm{I}_1 \text{ is } \mathrm{A}_1 \text{ and } \mathrm{I}_2 \text{ is } \mathrm{B}_1 \text{ and } \mathrm{I}_3 \text{ is } \mathrm{C}_3,\\
&\text{Then } f_{1,1,3} = \alpha_{1,1,3}\cdot \mathrm{I}_1 + \beta_{1,1,3}\cdot \mathrm{I}_2 + \gamma_{1,1,3}\cdot \mathrm{I}_3 + \eta_{1,1,3}\\
&\vdots\\
\text{Rule 27}: &\text{If } \mathrm{I}_1 \text{ is } \mathrm{A}_3 \text{ and } \mathrm{I}_2 \text{ is } \mathrm{B}_3 \text{ and } \mathrm{I}_3 \text{ is } \mathrm{C}_3,\\
&\text{Then } f_{3,3,3} = \alpha_{3,3,3}\cdot \mathrm{I}_1 + \beta_{3,3,3}\cdot \mathrm{I}_2 + \gamma_{3,3,3}\cdot \mathrm{I}_3 + \eta_{3,3,3}
\end{aligned} \tag{1}$$

where A_i, B_j, and C_k (i = 1 to 3) are the linguistic labels associated with this node function, respectively, i.e., the MFs for inputs I_i. $\alpha_{i,j,k}$, $\beta_{i,j,k}$, $\gamma_{i,j,k}$ and $\eta_{i,j,k}$ (i, j, k = 1 to 3) denote the consequent parameters. As shown in Fig. 2, the ANFIS's architecture is formed by using five layer and 27 (3^3) if-then rules as follows:

Layer 1

Each "i" node in this layer is a square node with a node function as

$$\begin{aligned}
O^1_{1,i} &= \mu_{A_i}(\mathrm{I}_1), \quad \text{for } i = 1,2,3\\
O^1_{2,j} &= \mu_{B_j}(\mathrm{I}_2), \quad \text{for } j = 1,2,3\\
O^1_{3,k} &= \mu_{C_k}(\mathrm{I}_3), \quad \text{for } k = 1,2,3
\end{aligned} \tag{2}$$

where I_1, I_2 and I_3 are inputs to node i, and $O^1_{1,i}$, $O^1_{2,j}$ and $O^1_{3,k}$ are the MFs of A_i, B_j, and C_k, respectively. The fuzzy MFs of $\mu_{A_i}(\mathrm{I}_1)$, $\mu_{A_i}(\mathrm{I}_1)$, and $\mu_{A_i}(\mathrm{I}_1)$ can be described in many types. In this study, four types of common MFs including Gaussian, generalized bell shaped, triangular and trapezoidal shaped functions with maximum value of 1 and minimum value of 0 were tested to find out the appropriate one.

Layer 2

In Layer 2, each circle node labeled Π multiplies the incoming signals and sends the product out. For instance,

$$O^2_{i,j,k} = w_{i,j,k} = \mu_{A_i}(\mathrm{I}_1)\cdot\mu_{B_j}(\mathrm{I}_2)\cdot\mu_{C_k}(\mathrm{I}_3), \quad i,j,k = 1,2,3 \tag{3}$$

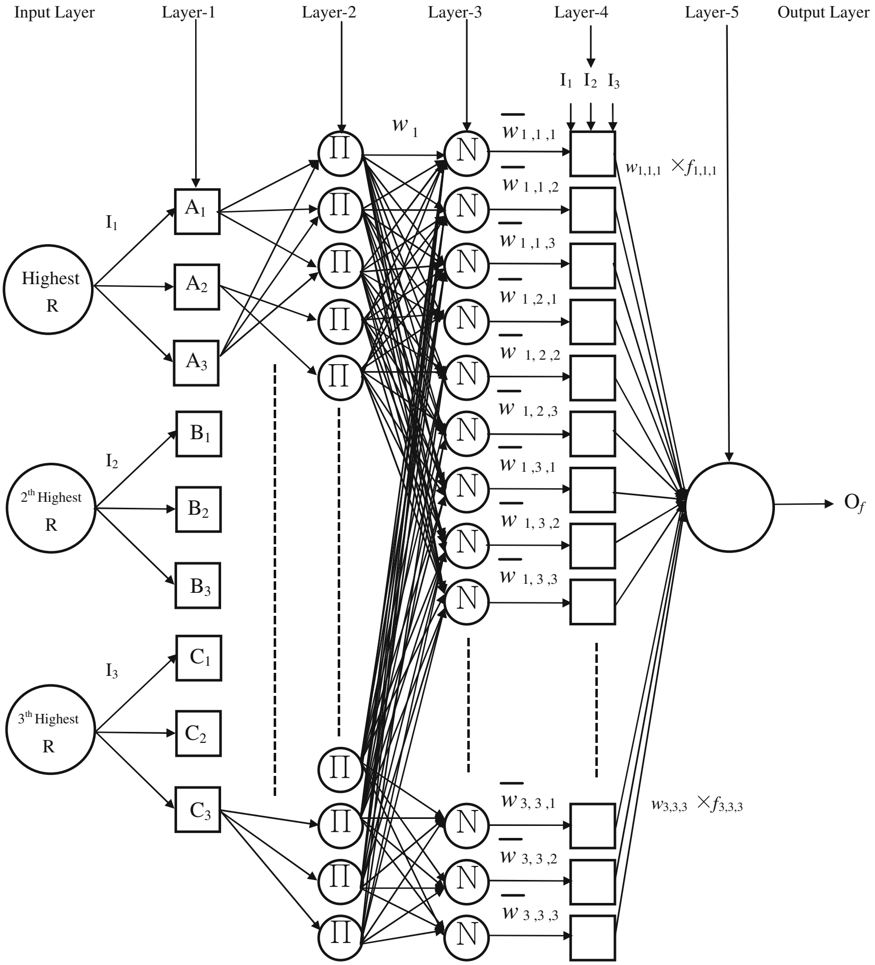


Fig. 2 ANFIS's architecture with 3 input variables and MFs [26]

Layer 3

In Layer 3, each circle node is labeled by N. The *i*th node calculates the ratio of the *i*th rule's firing strength to the sum of all rule's firing strengths, i.e., the normalized firing strength.

$$O^3_{i,j,k} = \overline{w}_{i,j,k} = \frac{w_{i,j,k}}{\sum_{i,j,k=1}^{3} w_{i,j,k}}, \quad i,j,k = 1,2,3 \tag{4}$$

Layer 4

Each square node i in this layer is a linear node function described as,

$$O^4_{i,j,k} = \overline{w}_{i,j,k} \cdot f_{i,j,k} = w_{i,j,k} \cdot \left(\alpha_{i,j,k}\mathrm{I}_1 + \beta_{i,j,k}\mathrm{I}_2 + \gamma_{i,j,k}\mathrm{I}_3 + \eta_{i,j,k}\right) i,j,k = 1,2,3 \quad (5)$$

Layer 5

The single circle node in this layer is depicted by Σ and computes the overall output as the summation of all incoming signals:

$$O_{5,i} = \text{Overall output } = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (6)$$

In the study proposed by Pai et al. [26], several types of ANFIS with different MFs and ANN were employed to predict hourly photochemical oxidants that were oxidizing substances such as ozone and peroxiacetyl nitrate produced by photochemical reactions. The results indicated that ANFIS statistically outperforms ANN in terms of hourly oxidant prediction. The minimum MAPEs of 4.99 % could be achieved using ANFIS with bell shaped MFs. The maximum correlation coefficient, the minimum mean square errors, and the minimum root mean square errors were 0.99, 0.15, and 0.39, respectively. ANFIS's architecture consists of both ANN and FL including linguistic expression of membership functions and if-then rules, so it can overcome the limitations of traditional ANN and increase the prediction performance.

3.2 Optimization

In addition to the improvement of structure, some newly developed biomimicry algorithms such as swarm intelligence was adopted to improve the prediction performance of ANN. Lu et al. [16, 17] adopted particle swarm optimization (PSO) to train perceptrons for prediction of air pollutant concentrations, and as a result, a PSO-based neural network approach was proposed. The approach was demonstrated to be effective and feasible by predicting some real air quality.

Wang and Lu [35] proposed a MLP model with a novel hybrid training method to forecast maximum daily ozone concentration in Hong Kong. The training approach combined a deterministic Levenberg–Marquardt (LM) algorithm and a stochastic PSO algorithm to exploit the advantage of both. Furthermore, the performance of the hybrid model was compared with those obtained by the MLP model. The simulation results showed that the hybrid model was more efficient than the other two models.

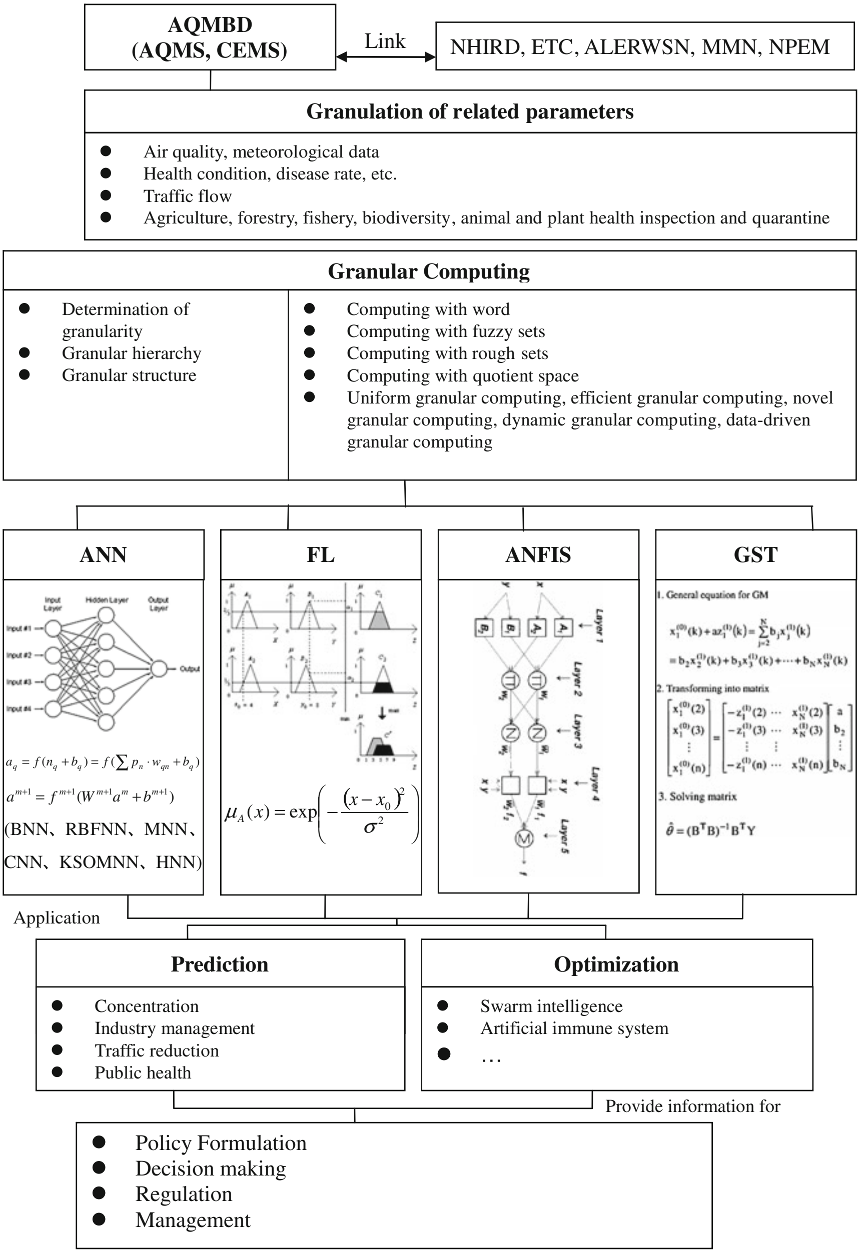

AQMBD
(AQMS, CEMS)
Link
NHIRD, ETC, ALERWSN, MMN, NPEM
Granulation of related parameters
Air quality, meteorological data
Health condition, disease rate, etc.
Traffic flow
Agriculture, forestry, fishery, biodiversity, animal and plant health inspection and quarantine
Granular Computing
Determination of granularity
Granular hierarchy
Granular structure
Computing with word
Computing with fuzzy sets
Computing with rough sets
Computing with quotient space
Uniform granular computing, efficient granular computing, novel granular computing, dynamic granular computing, data-driven granular computing
ANN
FL
ANFIS
GST
Input Layer
Hidden Layer
Output Layer
Input #1
Input #2
Input #3
Input #4
Output
$a_q = f(n_q + b_q) = f(\sum p_n \cdot w_{qn} + b_q)$
$a^{m+1} = f^{m+1}(W^{m+1}a^m + b^{m+1})$
(BNN、RBFNN、MNN、CNN、KSOMNN、HNN)
$\mu_A(x) = \exp\left(-\frac{(x-x_0)^2}{\sigma^2}\right)$
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
1. General equation for GM
$x_1^{(0)}(k) + az_1^{(1)}(k) = \sum_{j=2}^{N} b_j x_j^{(1)}(k)$
$= b_2 x_2^{(1)}(k) + b_3 x_3^{(1)}(k) + \cdots + b_N x_N^{(1)}(k)$
2. Transforming into matrix
3. Solving matrix
$\hat{\theta} = (B^T B)^{-1} B^T Y$
Application
Prediction
Concentration
Industry management
Traffic reduction
Public health
Optimization
Swarm intelligence
Artificial immune system
…
Provide information for
Policy Formulation
Decision making
Regulation
Management

Fig. 3 The flowchart for implementing CI on AQMBD granular computing

Although there is no application of granular computing on AQMBD at present, the flowchart for implementing CI on AQMBD granulating computing is suggested in Fig. 3.

4 Conclusions and Future Work

In this study, the formation and characteristics of AQMBD were investigated. It was found that the characteristics of AQMBD fit the properties of big data including volume, velocity, and variety (3Vs).

The application of CI on analysis of AQMBD was reviewed worldwide. The results revealed that various CI methods were used to predict the concentration of many types of air pollutants. Finally, the implementation of CI on AQMBD granular computing was proposed.

It was suggested that the application of CI and granular computing for analyzing AQMBD could be expanded in the future.

Acknowledgments The authors are grateful to the National Science Council of Taiwan, R.O.C. for financial support under the Grant Number NSC101-2621-M-142-001-MY2.

References

1. Atimtay, A.T., Emri, S., Bagci, T., Demir, A.U.: Urban CO exposure and its health effects on traffic policemen in Ankara. Environ. Res. **82**(3), 222–230 (2000)
2. Bargiela, A., Pedrycz, W.: Granular Computing: an Introduction. Kluwer Academic Publishers, Boston (2002)
3. Boznar, M., Lesjak, M., Mlakar, P.: A neural network-based method for short-term predictions of ambient SO_2 concentrations in highly polluted industrial areas of complex terrain. Atmos. Environ. Part B **27**(2), 221–230 (1993)
4. Cunningham, W.P., Cunningham, M.A.: Principles of Environmental Science Inquiry & Applications. McGraw-Hill Company, New York (2008)
5. Cai, M., Yin, Y., Xie, M.: Prediction of hourly air pollutant concentrations near urban arterials using artificial neural network approach. Transp. Res. D-Tr E **14**(1), 32–41 (2009)
6. Chen, S.M., Yang, M.W., Lee, L.W., Yang, S.W.: Fuzzy multiple attributes group decision-making based on ranking interval type-2 fuzzy sets. Expert Syst. Appl. **39**(5), 5295–5308 (2012)
7. Comrie, A.C.: Comparing neural networks and regression models for ozone forecasting. J. Air Waste Manage. Assoc. **47**, 653–663 (1997)
8. De Kok, T.M.C.M., Driece, H.A.L., Hogervorst, J.G.F., Briedé, J.J.: Toxicological assessment of ambient and traffic-related particulate matter: a review of recent studies. Mutat. Res. **613**(2–3), 103–122 (2006)
9. Delfino, R.J., Murphy-Moulton, A.M., Becklake, M.R.: Emergency room visits for respiratory illnesses among the elderly in Montreal: association with low level ozone exposure. Environ. Res. **76**, 67–77 (1998)
10. Finkelstein, M.M., Jerrett, M.: A study of the relationships between Parkinson's disease and markers of traffic-derived and environmental manganese air pollution in two Canadian cities. Environ. Res. **104**(3), 420–432 (2007)
11. Gardner, M.W., Dorling, S.R.: Artificial neural networks (the multilayer feed-forward neural networks)—a review of applications in the atmospheric science. Atmos. Environ. **30**(14/15), 2627–2636 (1998)
12. Gautam, A.K., Chelani, A.B., Jain, V.K., Devotta, S.: A new scheme to predict chaotic time series of air pollutant concentrations using artificial neural network and nearest neighbor searching. Atmos. Environ. **42**(18), 4409–4417 (2008)

13. Hyder, A.A., Ghaffar, A.A., Sugerman, D.E., Masood, T.I., Ali, L.: Health and road transport in Pakistan. Publ. Health **120**(2), 132–141 (2006)
14. Lee, E., Chan, C.K., Paatero, P.: Application of positive matrix factorization in source apportionment of particulate pollutants. Atmos. Environ. **33**, 3201–3212 (1999)
15. Lu, W.Z., Wang, W.J.: Potential assessment of the "support vector machine" method in forecasting ambient air pollutant trends. Chemosphere **59**, 693–701 (2005)
16. Lu, W.Z., Fan, H.Y., Leung, A.Y.T., Wong, J.C.K.: Analysis of pollutant levels in central Hong Kong applying neural network method with particle swarm optimization. Environ. Monit. Assess. **79**(3), 217–230 (2002)
17. Lu, W.Z., Fan, H.Y., Lo, S.M.: Application of evolutionary neural network method in predicting pollutant levels in downtown area of Hong Kong. Neurocomputing **51**, 387–400 (2003)
18. Lu, W.Z., Wang, W.J., Wang, X.K., Xu, Z.B., Leung, A.Y.T.: Using improved neural network model to analyze RSP, NO_x and NO_2 levels in urban air in Mong Kok. Hong Kong. Environ. Monit. Assess. **87**(3), 235–254 (2003)
19. Pai, T.Y., Hanaki, K., Ho, H.H., Hsieh, C.M.: Using grey system theory to evaluate transportation on air quality trends in Japan. Transp. Res. D-Tr E **12**(3), 158–166 (2007)
20. Pai, T.Y., Tsai, Y.P., Lo, H.M., Tsai, C.H., Lin, C.Y.: Grey and neural network prediction of suspended solids and chemical oxygen demand in hospital wastewater treatment plant effluent. Comput. Chem. Eng. **31**(10), 1272–1281 (2007)
21. Pai, T.Y., Chuang, S.H., Ho, H.H., Yu, L.F., Su, H.C., Hu, H.C.: Predicting performance of grey and neural network in industrial effluent using online monitoring parameters. Process Biochem. **43**(2), 199–205 (2008)
22. Pai, T.Y., Chuang, S.H., Wan, T.J., Lo, H.M., Tsai, Y.P., Su, H.C., Yu, L.F., Hu, H.C., Sung, P.J.: Comparisons of grey and neural network prediction of industrial park wastewater effluent using influent quality and online monitoring parameters. Environ. Monit. Assess. **146**(1–3), 51–66 (2008)
23. Pai, T.Y., Lin, K.L., Shie, J.L., Chang, T.C., Chen, B.Y.: Predicting the co-melting temperatures of municipal solid waste incinerator fly ash and sewage sludge ash using grey model and neural network. Waste Manage. Res. **29**(3), 284–293 (2011)
24. Pai, T.Y., Yang, P.Y., Wang, S.C., Lo, H.M., Chiang, C.F., Kuo, J.L., Chu, H.H., Su, H.C., Yu, L.F., Hu, H.C., Chang, Y.H.: Predicting effluent from the wastewater treatment plant of industrial park based on fuzzy network and influent quality. Appl. Math. Model. **35**(8), 3674–3684 (2011)
25. Pai, T.Y., Ho, C.L., Chen, S.W., Lo, H.M., Sung, P.J., Lin, S.W., Lai, W.J., Tseng, S.C., Ciou, S.P., Kuo, J.L., Kao, J.T.: Using seven types of GM (1, 1) model to forecast hourly particulate matter concentration in Banciao City of Taiwan. Water Air Soil Pollut. **217**(1–4), 25–33 (2011)
26. Pai, T.Y., Hanaki, K., Su, H.C., Yu, L.F.: A 24-h forecast of oxidant concentration in Tokyo using neural network and fuzzy learning approach. Clean-Soil, Air, Water **41**(8), 729–736 (2013)
27. Pedrycz, A., Hirota, K., Pedrycz, W.: Fangyan Dong: Granular representation and granular computing with fuzzy sets. Fuzzy Sets Syst. **203**, 17–32 (2012)
28. Pedrycz, W., Bargiela, A.: An optimization of allocation of in-formation granularity in the interpretation of data structures: toward granular fuzzy clustering. IEEE Trans Syst. Man Cybern. B Cybern. **42**(3), pp. 582–590 (2012)
29. Pedrycz, W., Song, M.: Granular fuzzy models: a study in knowledge management in fuzzy modeling. Int. J. Approx. Reason. **53**(7), 1061–1079 (2012)
30. Pedrycz, W., Homenda, W.: Building the fundamentals of granular computing: a principle of justifiable granularity. Appl. Soft Comput. **13**(10), 4209–4218 (2013)
31. Perez, P., Trier, A., Reyes, J.: Prediction of PM2.5 concentrations several hours in advance using neural networks in Santiago Chile. Atmos. Environ. **34**, 1189–1196 (2000)
32. Reich, S.L., Gomez, D.R., Dawidowski, L.E.: Artificial neural network for the identification of unknown air pollution sources. Atmos. Environ. **33**, 3045–3052 (1999)

33. Roadknight, M., Balls, G.R., Mills, G.E., Palmer-Brown, B.D.: Modelling complex environmental data. IEEE Trans Neural Netw. **8**(4), pp. 852–861 (1997)
34. Song, X.H., Hopke, P.K.: Solving the chemical mass balance problem using an artificial neural network. Environ. Sci. Technol. **30**(2), 531–535 (1996)
35. Wang, D., Lu, W.Z.: Forecasting of ozone level in time series using MLP model with a novel hybrid training algorithm. Atmos. Environ. **40**(5), 913–924 (2006)
36. Wang, W., Lu, W., Wang, X., Leung, A.Y.T.: Prediction of maximum daily ozone level using combined neural network and statistical characteristics. Environ. Int. **29**(5), 555–562 (2003)
37. Wang, W., Xu, Z., Lu, W.: Three improved neural network models for air quality forecasting. Eng. Computation **20**(2), 192–210 (2003)
38. Yi, J., Prybutok, R.: A neural network model forecasting for prediction of daily maximum ozone concentration in an industrialized urban area. Environ. Pollut. **92**(3), 349–357 (1996)
39. Zadeh, L.A.: Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets Syst. **90**(2), 111–127 (1997)
40. Zadeh, L.A., Gupta, M., Ragade, R.K., Yager, R.R. (eds.): Fuzzy Sets and Information Granulation, Advances in Fuzzy Set Theory and Applications. North-Holland Publishing Company, Amsterdam (1979)
41. Zadeh, L.A.: Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. Soft. Comput. **2**(1), 23–25 (1998)
42. Zhang, B., Zhang, L.: Theory and Application of Problem Solving. North Holland (1992)

Index

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7,