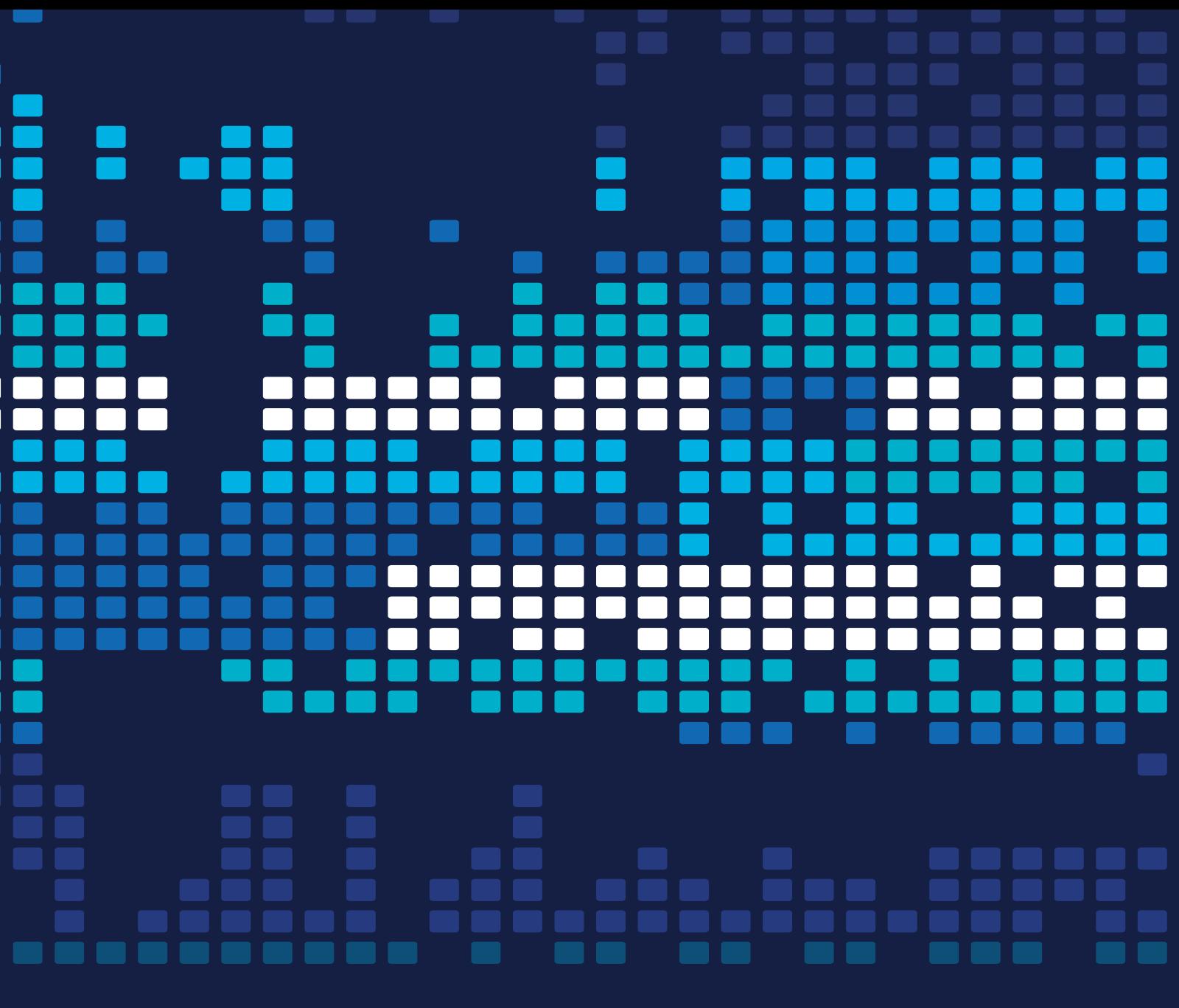
Scientific Programming

Programming Foundations for Scientific Big Data Analytics

Lead Guest Editor: Wenbing Zhao
Guest Editors: Longxiang Gao and Anfeng Liu



Programming Foundations for Scientific Big Data Analytics

Scientific Programming

Programming Foundations for Scientific Big Data Analytics

Lead Guest Editor: Wenbing Zhao
Guest Editors: Longxiang Gao and Anfeng Liu

Editorial Board

Contents

Hindawi
Scientific Programming
Volume 2018, Article ID 2707604, 2 pages
https://doi.org/10.1155/2018/2707604

Editorial

Programming Foundations for Scientific Big Data Analytics

Wenbing Zhao,[1] **Longxiang Gao**,[2] **and Anfeng Liu**[3]

[1]*Cleveland State University, Cleveland, OH, USA*
[2]*Deakin University, Burwood, NSW, Australia*
[3]*Central South University, Changsha, China*

Correspondence should be addressed to Wenbing Zhao; w.zhao1@csuohio.edu

Received 7 March 2018; Accepted 7 March 2018; Published 19 April 2018

Big data analytics is the process of examining large data sets to uncover hidden patterns and previously unknown correlations. Big data analytics has been widely used in businesses to find market trends, customer preferences, and other useful business information. The research community is also beginning to embrace this exciting and powerful technology. Considering the huge amount of data produced in scientific fields such as biology, medicine, physics, and material science, big data analytics can be a powerful means of making new scientific discoveries. Efficient and effective big data analytics requires the development of programming tools and models.

This special issue attracted 20 high quality submissions. After a rigorous review process, 13 papers were accepted in this issue. The research presented in these papers can be roughly categorized into three areas: (1) platform for big data analytics (3 papers), (2) machine learning algorithms for big data (6 papers), and (3) big data analytics for various applications (4 papers).

Platform for Big Data Analytics. B. R. Chang et al. reported their work on how to integrate popular big data platforms such as Hadoop and Spark to perform high performance big data analytics. They focused on the optimization of job scheduling based on computing features to improve system throughout. L. Zhang and J. Gao introduced a novel incremental graph pattern matching algorithm for big graph data. By batching insert operations together by considering matching states, they were able to demonstrate higher efficiency of the proposed algorithm. L. Yang et al. proposed several optimization algorithms based on node compression to help solve the shortest path problem in the context of routing big data.

Machine Learning Algorithms for Big Data. H.-F. Ke et al. proposed a new optimal weight learning machine that is capable of incremental learning while the network grows in terms of the number of hidden nodes. L. Wang et al. presented an adaptive ensemble method for classification with imbalanced data. They rely on self-adaption based on the average Euclidean distance between test data and training data, which is obtained by the k-nearest neighbors algorithm. W. Aziguli et al. introduced a new algorithm designed specifically for text classification, which could be useful for analyzing text-based big data. The algorithm is based on the use of denoising autoencoder and restricted Boltzmann machine, which has the advantage of better performance in antinoise and feature extraction. Z. Yuan et al. proposed a new matching pursuit method to overcome the singularity problem and improve the stability of extreme learning machine (ELM).

X. Fan et al. reported a new hybrid similarity calculation model, which is essential in many machine learning algorithms. The model was designed specifically for recommendation algorithms by addressing the user interest drift issue. This model uses the function fitting to reflect users' rating behaviors and their rating preferences and employs the Random Forest algorithm for the user attribute features. X. Pu et al. proposed a hybrid biogeography-based optimization algorithm for big data analytics. The algorithm is used with a feedforward neural network model called multilayer perceptron.

Big Data Analytics for Various Applications. Y. Zhu et al. reviewed the latest development on big data management in the field geological information services. They proposed a system architecture and outlined requirements for big data management for this application domain. D. Li et al. proposed

a loss aversion cooperation model for behavioral economics in crowd-sensing. They showed that their model encourages higher cooperation rate with lower pay rate.

S. Zhang et al. introduced a new algorithm to make prediction on advertisement click-through rate. The algorithm is based on the weighted extreme learning machine and the Adaboost algorithm. A more accurate predication on click-through rate would increase advertising performance, which could lead to the improvement of an advertising company's reputation and revenue.

X. Xia et al. reported their work on developing a monitoring and prewarning system for accidents in the coal mines using data collected by a network of wireless sensors. They proposed a new data aggregation strategy and fuzzy comprehensive assessment model to derive useful information based on the collected data.

Acknowledgments

The guest editors would like to thank the authors for contributing to this special issue and thank all the reviewers for their time and rigorous reviews.

Wenbing Zhao
Longxiang Gao
Anfeng Liu

Hindawi
Scientific Programming
Volume 2018, Article ID 4350183, 18 pages
https://doi.org/10.1155/2018/4350183

Research Article

Analysis of Behavioral Economics in Crowdsensing: A Loss Aversion Cooperation Model

Deng Li,[1] Liying Qiu,[1] Jiaqi Liu ,[2] and Congwen Xiao[3]

[1]*School of Information Science and Engineering, Central South University, Changsha 410083, China*
[2]*School of Software, Central South University, Changsha 410083, China*
[3]*School of Computer and Communication Engineering, Northeastern University, Qinhuangdao 066000, China*

Correspondence should be addressed to Jiaqi Liu; liujiaqi@csu.edu.cn

Received 27 October 2017; Revised 6 February 2018; Accepted 19 February 2018; Published 12 April 2018

Academic Editor: Longxiang Gao

The existing incentive mechanisms of crowdsourcing construct the expected utility function based on the assumption of rational people in traditional economics. A large number of studies in behavioral economics have demonstrated the defects of the traditional utility function and introduced a new parameter called loss aversion coefficient to calculate individual utility when it suffers a loss. In this paper, combination of behavioral economics and a payment algorithm based on the loss aversion is proposed. Compared with usual incentive mechanisms, the node utility function is redefined by the loss aversion characteristic of the node. Experimental results show that the proposed algorithm can get a higher rate of cooperation with a lower payment price and has good scalability compared with the traditional incentive mechanism.

1. Introduction

Crowdsensing, which can be described as people/human-centric sensing, has gradually become the ideal method for large-scale data collection [1]. The collection of sensed data relies on every single node that participates in the perception. Each node is a natural person holding smart devices such as smart phones or computers. Crowdsensing systems usually require support from a large amount of sensed data [2–5], while it would cost smart devices a certain amount of price to participate in crowdsensing (such as spatial movements and consumption of memory and power). Nodes do not perceive selflessly, as they need a certain amount of compensation to be motivated; on the other hand, the requester of the sensing tasks would measure the reward of publishing the value of data. A sensing task usually offers limited reward, so it would be better to get more valuable sensed data with less reward.

At present, there are a series of incentive mechanisms, which seek benefit maximization mainly based on utility function, which is based on the expected function theory of traditional economics. However, the development of economics itself has gradually revised this theory. The behavioral economics shows that individual decision-making must consider the influence of psychological factors. The traditional expected utility function is no longer completely reasonable when psychological-related parameters are involved.

The loss aversion is an important branch of the prospect theory of behavioral economics. What it describes is loss is more unbearable than gain that has the same amount of value. Therefore, individuals are more willing to voluntarily prevent possible losses. In order to motivate the nodes, this paper, from the point of view of the nodes, analyzes the decision-making process of the nodes, models them, introduces the loss aversion coefficient into the utility function of them, adjusts the payment mode in the traditional incentive mechanisms, effectively stimulates nodes to participate in perception, and enhances the performance of the crowdsensing system.

The main innovations of this paper are as follows:

(i) The paper uses the loss aversion to build the incentive mechanism, which revises the cooperative behavior researches based on traditional economics, so as to make up for the basic assumption insufficiency in the

traditional economics about human rationality, self-interest, complete information, utility maximization, and preference consistent.

(ii) By using the influence of the loss aversion psychology on decision-making, a compensation payment algorithm based on the loss aversion is proposed in the crowdsensing.

2. Related Work

2.1. The Incentive Mechanisms of Crowdsensing. In order to increase the number of participants in sensing tasks and ensure the data quality, a series of incentive mechanisms have been put forward, including monetary incentive, entertainment and gamification incentive, and social connections incentive [6]. References [7, 8] pointed out that the attributes of human beings are diversified and individuals' decision-making behaviors are influenced not only by their own cognitive, thinking, preferences, and other factors, but also by the surrounding environment at the same time. According to the motivation mechanism, this subject makes use of the individual's individuality and sociality and divides the common incentive mechanisms into individual incentive mechanisms and social incentive mechanisms.

(1) Individual Incentive Mechanisms. The individual incentive mechanisms mainly utilize the inherent pursuit of interests of the nodes, including the desire for money, the motivation to maintain and manage its own reputation, the pursuit of more virtual points, and a better entertainment experience.

(i) Mechanisms Based on Monetary Payment. The incentive mechanisms based on the monetary payment are that the platform motivates potential participants to join the sensing task and provides the required sensed data by giving workers a certain monetary reward. This kind of incentive mechanisms is usually the combination of economics and computer science. The most common auction mechanisms include reverse auctions [9–12], portfolio auctions [13], multiattribute auctions [14], all pay auctions [15], double auctions [16], and VCG (Vickrey-Clarke-Groves) auctions [17]. In monetary incentive mechanisms, game-theoretic mechanisms provide good mathematical models to resolve server-to-player conflicts and determine problems, while providing sufficient theoretical data to analyze participants' behavior. The monetary incentive mechanism can effectively stimulate the enthusiasm of participants [18] and has a good theoretical basis. However, it also has obvious shortcomings. The system usually can hardly establish a suitable price architecture. Most importantly, the current pricing scheme cannot solve the dilemma between the requesters and the workers: If paid in advance, the workers can get the reward without working, which is called free-riding; if paid afterwards, the requesters can refuse to pay after getting the required information, which is called false-reporting [19].

(ii) Mechanisms Based on Entertainment and Gamification. The incentive mechanisms based on entertainment and gamification change the sensing tasks to sensing games, so as to allow workers to contribute to the sensing tasks in the game process. The mechanisms usually motivate workers to complete the sensing tasks by generating rankings in the game, task points and their intrinsic fun, and so on. The authors in [20] used a ranking scheme and a badge scheme to motivate workers to participate in. The authors in [21] designed a collection game called Treasure to collect information in the gaming area to draw a Wi-Fi coverage map. The author in [22] used a player's text or photo tag in the play area to generate a series of recognizable geographic information that supports route navigation.

Individual motivation mechanisms neglect the environment in which the nodes are located. In the crowdsensing, individual nodes have the ability to interact with other nodes, and their behaviors and connections are mutually influential. Some researchers found that, in the incentive mechanisms of crowdsensing, the position of the workers' social structure will affect their resources and access to information, as well as the degree of completion of the sensing tasks and the amount of needed compensation [23]. Therefore, a series of social incentive mechanisms for nodes are proposed.

(2) Social Incentive Mechanisms. Social incentive mechanisms consider the social aspects of nodes. Crowdsensing is made up of a large number of nodes, so the choices and behaviors among nodes are not completely isolated. Nodes adjust self-cognition at any time by the influence of other nodes and they draw up their behavioral strategy based on the information in social networks.

(i) Mechanisms Based on Social Connections. The incentives mechanisms based on social connections focus on the interactions and relationships among individuals. The authors in [24] established the social network among the participants based on Stackelberg, in order to maximize the participants' utility. Based on social networks, the authors in [25] use a penalty mechanism to detect dishonest participants and build a trust system to improve existing incentive mechanisms. The authors in [26] improve the choice of participants and payment of remuneration, to enhance the integrity of the individual by supervision and reporting in nepotism. Incentive mechanisms based on social connections motivate the participants to a certain extent, but because of the impact of network relations itself, the reliability and credibility of social networks are in bad need of [18].

(ii) Mechanisms Based on Service. The incentive mechanisms based on service are designed by using the principle of mutual benefits. In the crowdsensing, service consumers can also be considered as service providers. That is, if the nodes want to get the services provided by the system, the nodes must also contribute to the system. For example, in the Parking Information System designed by [27], nodes play the roles of both consumers and contributors. The authors in [28] have designed two incentive mechanisms under this framework: Incentive with Demand Fairness (IDF) and Iterative Tank Filling (ITF). They are used separately to maximize the fairness of nodes and the social welfare of the system. Some researchers consider service incentives from a group level.

The authors in [29] illustrate the inspiration from blood donation that contributors are driven not only by their own utility but also by the effects of their relatives and friends. And this group incentive has proved to be effective in practice.

The above incentive mechanisms based on the utility are put forward to maximize utilities of both the platform and the participants. The model of those incentive mechanisms can be expressed as the following formula [30]:

$$I : M \longrightarrow \max\left(U\left(S\right), U\left(P\right)\right). \tag{1}$$

That is reflected in the classic monetary incentive mechanisms [9–17]: each node makes its decisions to maximize its payment with the lowest cost. All the nodes, during the bidding between the server and them, would maintain their rationality for more benefits. In entertainment and gamification incentive mechanisms [20–22], virtual credits and ranks take the place of money, so that nodes would make decisions that maximize their interests. Moreover, considering the interactions within the nodes, social incentive mechanisms aim at maximizing utilities of groups instead of individuals [18, 24–29].

All in all, as shown in formula (2), most of incentive mechanisms still use the traditional expected utility function to describe the decision-making of the individuals:

$$U\left(S\right) = \max E\left[U\right] = \sum_{i} p_i U\left(x_i\right). \tag{2}$$

Formula (2) is used to describe the utility U of a node S, taking p_i as the probability for choosing event x_i and $U(x_i)$ as the utility of event x_i (the value of $U(x_i)$ could be positive or negative). Each individual would calculate its expected utility according to formula (2) and use the result for its decision-making.

Formula (2) is based on the two hypotheses from traditional economics: source independence and the invariance of risk preference [31].

Source independence means the fungibility of wealth. In traditional economics theory, the values of wealth do not depend on how it is acquired, nor are they labeled [32]. That is, nodes measure their gains and losses in a similar way. The differences between benefits and costs could be circulated, which is obvious in the mechanisms which evolved punishment [25, 26]: the cost of punishment and the benefit of cooperation could be superimposed without any differences.

The invariance of risk preference means that the risk preference of the individuals is constant, objective, and consistent, which has been called process invariance in traditional economics [33]. In this situation, individuals in crowdsensing would never change their risk preference, no matter how the information environment in the system changes. In the mechanisms of social networks, individuals attitudes towards risk of pursuing benefits or avoiding losses are constant, although they would change their decisions according to their opponents.

However, incentive mechanisms apart from monetary incentive mechanisms [9–17] (e.g., entertainment and gamification incentive mechanisms [20–22] and social incentive mechanisms [18, 24–26]) consider not only the economic gains and losses, but also the psychological factors of the individuals. So it is not reasonable to calculate individuals' utilities with the methods from monetary incentive mechanisms. Even in the monetary incentive mechanisms, the impacts of some factors (such as the values of money, the sources of money, and the risk tendencies of money) on decision-making cannot be ignored. In addition, the two hypotheses including the independence of sources and the invariance of risk preference have been questioned [34]. A number of studies have demonstrated that individuals' irrational behaviors may be refracted into the program and that will lead to the occurrence of the irrational decisions [35].

2.2. Incentive Mechanism from the Perspective of Behavioral Economics. Behavioral economics put forward a well-known theory called loss aversion, which states that losses are even more unbearable when people face the same amount of benefits and losses [36]. The loss aversion has been proved to be a common feature embodied in individual decisions.

The loss aversion has changed the expected utility function in traditional economics and introduced an unprecedented loss aversion coefficient λ to describe the loss aversion characteristics of individuals, which is used to calculate the utility of individuals when they suffer losses as the following formula [37]:

$$U\left(x\right) = \begin{cases} x, & x \gg 0 \\ \lambda x, & x < 0 \end{cases} \quad \lambda > 1. \tag{3}$$

Formula (3) is a simplified form of the loss-aversion-typed value function, with 0 as the reference point to express the individual gains and losses. After introducing the loss aversion coefficient, this function overthrows the traditional expected utility function and can be used to explain a series of phenomena that cannot be explained by the expected utility function, such as Allais paradox [38], reflex effect [39], and preference reversal [40].

Therefore, the actual individual decision-making must consider the utility function after the loss aversion coefficient is considered.

$$U'\left(x_i\right) = \begin{cases} U\left(x_i\right), & U\left(x_i\right) \gg 0 \\ \lambda U\left(x_i\right), & U\left(x_i\right) < 0 \end{cases} \quad \lambda > 1$$
$$U'\left(S\right) = \sum_{i} p_i U\left(x_i\right) \neq \max E\left[U\right]. \tag{4}$$

Formula (4) adds that when the utility $U(x_i)$ corresponding to event x_i is negative, its value changes under the influence of the loss aversion coefficient, which will affect the overall utility function.

Besides, the loss aversion also overturns the view of source independence and the invariance of risk preference

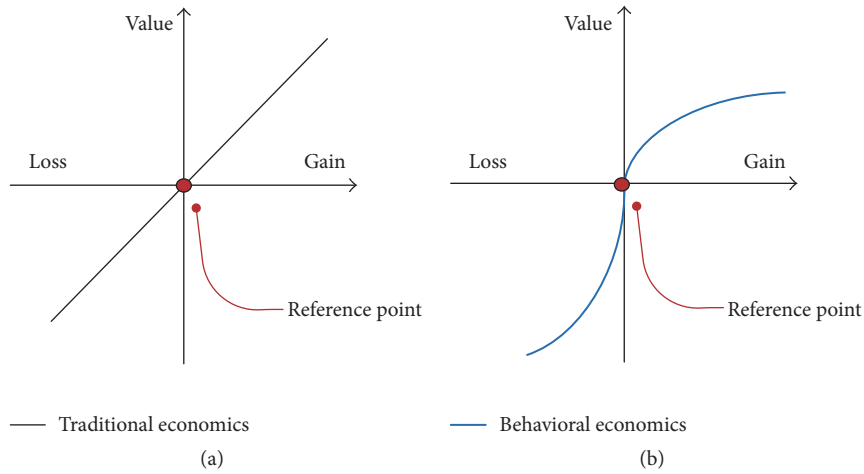


Figure 1: The value curve of traditional economics versus behavioral economics.

in traditional economics and considers that money has nonfungibility and individual risk preference is variable [41].

(1) Nonfungibility. Being different from the view of source independence, behavioral economics holds that when external information enters the individual cognitive mental accounting, it can effectively reflect the trade-off between expected return and possible loss and confirm whether the threshold value boundary has been reached [42]. Because of this effective boundary, money cannot flow freely among subaccounts. In other words, due to the existence of the loss aversion, when individuals make decision, the value of funds is different and irreplaceable according to different sources and expenditures. The value curve of traditional economics is like Figure 1(a), and the actual value function should be like Figure 1(b).

(2) The Variability of Risk Preference. Individual's risk preference plays an important guiding role in individual decision-making. Behavioral economics considers that the loss aversion leads to the change of individuals' risk preference. Individuals are risk-averse when faced to defined returns, while they tend to seek risks when faced to established losses [43]. When making economic decisions, individuals are more worried about losing money than expecting to gain profits, which in turn encourages them to have more motivation to stop the loss. This is also a sign of the loss aversion. The crowdsensing system is fuzzy and uncertain for nodes. In addition to considering the proceeds, the nodes in the system are more worried about their loss than other nodes, which will greatly affect the decision-making behavior of the nodes.

To sum up, the crowdsensing system is an environment with unequal information, uncertainty, and ambiguity. Nodes are inevitably affected by this environment and cannot maintain individual rationality, and utility function is bound to be influenced by psychological factors. Because the loss aversion theory has been well applied in the fields of economics [44], game theory [45], biology [46], environmental ecology [47], and so on, it has proved its feasibility. However, according to our study, the loss aversion has not been used in the field of crowdsensing, so we believe that the incentive mechanism of the past crowdsensing system did not make full use of the characteristics of the node and target the incentives. After the loss aversion is introduced, the structure of the incentive mechanism can be extended to formula (5), where $E(S)$ represents the psychological factors of the nodes:

$$I : M \longrightarrow \max\left(U\left(S\right), U\left(P\right), E\left(S\right)\right). \tag{5}$$

3. Our Mechanism

There is a famous grape experiment [48] to verify the loss aversion, which clearly shows how the loss aversion works in the psychological aspects. We have established the mapping of the experiment and the crowdsensing system. We propose a different payment algorithm by the reasonable analysis and construction of the model. In this algorithm, the nodes' loss aversion is aroused, so as to be more proactive in the perception to improve system efficiency.

3.1. The Introduction of the Loss Aversion. Monkeys were used as subjects in the grape experiment, and the experimenter designed two game schemes for monkeys:

(i) Option 1: the experimenter first puts a grape in front of the a monkey and does a coin toss. The coins face down, allowing the monkey to only get that one grape; the coins face up, then allowing the monkey to get an extra grape. The expectation that each monkey can get grapes is 1 + 50%.

(ii) Option 2: the experimenter places two grapes in front of a monkey and then does a coin toss. The coins face down; then the experimenter takes one of the grapes; the coins face up; then the monkey can get both the two grapes. The expectation that each monkey can get grapes is 2–50%.

For the sake of reason, the expected benefits of both experiments are 1.5 grapes. The difference is that, in the first

Table 1: Mapping of the grape experiment and incentive mechanisms.

	Participants	Step one	Step two	Regulation
Grape experiment option 1	Monkeys	Put one grape in front of the monkeys	Do a coin toss	Coins face up; add 1 grape
Grape experiment option 2	Monkeys	put two grapes in front of the monkeys	Do a coin toss	Coins face down; take away 1 grape
Traditional incentive mechanism	Network nodes	The node chooses whether to participate	The node chooses whether to participate	Increase the node contribution c_2 only if the node cooperates
New incentive mechanism based on the loss aversion	Network nodes	Increase the node contribution $c_1 + c_2$ to the capable nodes	The node chooses whether to participate	Decrease the node contribution c_2 only if the node does not cooperate
Common	Having a certain understanding to determine their own interests	Encourage monkeys to participate in experiment; encourage nodes to join in system	Although the coin toss is a probability event, the choice of cooperation is the node's own decision; they are for the subsequent results and monkeys/nodes know how it works	Additional gained contribution (grape) and feeling happy, or loss of contribution (grape) and feeling pain

option, each monkey has 50% chance to get an extra grape in the case of ensuring a grape is obtained, while the second option is 50% chance to lose one of the grapes on the premise that two grapes may be obtained.

If the monkey is a completely rational individual, then its preference for these two experiments should be the same. However, the experimental results show that when the monkeys finally understand they may lose one of the two grapes in option 2, they all tend to choose option 1. This experiment shows that the pain brought by the loss of a grape to the monkeys is heavier than the happiness brought by getting a grape.

We simulate the grape experiment and propose a new mechanism for enhanced cooperation based on the loss aversion, which is different from the traditional incentive mechanisms, as shown in Table 1.

Theorem 1. *The individual's pain of loss is often greater than the value of the actual loss when measuring the gains and losses, expressed as follows:*

$$c_2\big|_{\text{lost}} > c_2\big|_{\text{gain}}. \tag{6}$$

Proof. Traditional economics argues that the resulting c_2 is equal in value to the lost c_2, expressed as follows:

$$c_2\big|_{\text{gain}} = c_2\big|_{\text{lost}} = |c_2|. \tag{7}$$

However, due to the impact of loss of aversion on the value curve, the individual really perceived that loss part of the value should be adjusted as follows:

$$c_2'\big|_{\text{lost}} = \lambda |c_2|^{\beta}. \tag{8}$$

Since

$$c_2'\big|_{\text{lost}} - c_2\big|_{\text{gain}} = \lambda |c_2|^{\beta} - |c_2| = \left(\lambda - |c_2|^{1/\beta}\right)|c_2|^{\beta} \tag{9}$$

$|c_2|^{\beta} > 0$, so when $\lambda > |c_2|^{1/\beta}$, which is $|c_2| > \log_{\beta}(1/\lambda)$, we can get $c_2'|_{\text{lost}} > c_2|_{\text{gain}}$. That is,

$$c_2'\big|_{\text{lost}} > c_2\big|_{\text{gain}} \quad \text{when } |c_2| > \log_{\beta}\frac{1}{\lambda}. \tag{10}$$

□

Based on the above analysis, the new mechanism of the loss aversion is to enlarge the value of c_2 in the psychological level, so that limited rational nodes tend to choose the cooperation without losing the contribution value, thus promoting the enthusiasm of the node.

3.2. System Modeling. We use the classic crowdsensing architecture to build the system. The typical system architecture is shown in Figure 2. The system includes the server platform and task participants (data providers). The server in the cloud receives a service request from the data requesters (the data requester can be the data providers; they are the same group), assigns the sensing task to the participants, processes the collected sensed data, and performs other administrative functions. After a participant receives the sensing task, the participant senses the required data and then uploads the data to the server. The server returns the data to the data requesters after processing.

(i) The finite set of $T = \{t_1, t_2, t_3, \ldots, t_n\}$ represents the service requests from the data users, that is, the sensing tasks that need to be completed in the system.

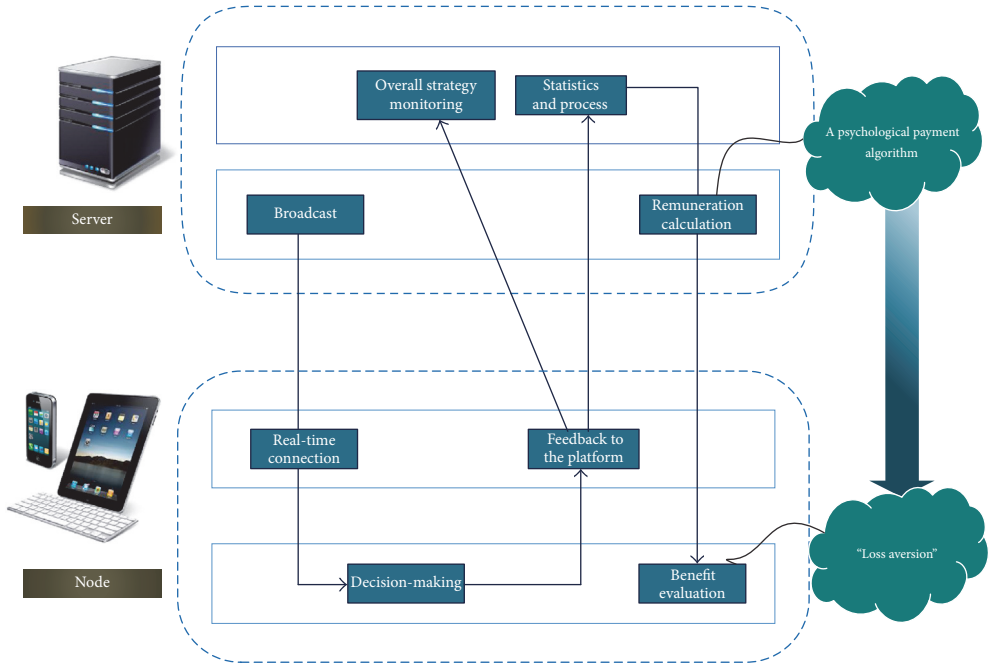


Figure 2: Typical system architecture of crowdsensing.

(ii) $N = \{1, 2, 3, \ldots, n\}$ represents the number of workers of the system; these workers are handheld intelligent devices in practical applications. A worker can only join one sensing process at a time.

3.2.1. Platform. Each of the sensing tasks t_i $(i \in N)$ has its own attributes. We extract the main attributes needed in this paper, denoted as $\langle P(t_i), B(t_i), \mathrm{Con}(ti), t_i \in T\rangle$, $P(t_i)$ means its payoff, $B(t_i)$ means its budget, and $\mathrm{Con}(ti)$ means its congestion level, respectively, and congestion level indicates the current degree of participation of the task.

There are various reward pricing schemes in the crowdsensing system. A budget-limited perception task has a rated total remuneration. In order to obtain plenty of high-quality sensed data in a defined amount, this paper defines that each sensing task is paid to each worker with a reward of $P(t_i) = B(t_i)/\mathrm{Con}(ti), \mathrm{Con}(ti) = |\{n \in N : s_n = t_i\}|$, where s_n represents the current strategy of a worker N.

In some cases, a worker may refuse to provide sensed data, and we define this behavior as sleeping in which the node does not have to pay any perceived costs and will not receive any remuneration. We manually introduce t_0 to describe it; t_0's reward and the degree of congestion are always zero. The expanded task set is $\mathcal{T} = \{t_0, t_1, t_2, t_3, \ldots, t_n\}$.

3.2.2. Worker. We describe each worker as a quaternion $\langle G_n^t, C_n^t, A_n^t, AI_n^t, n \in N\rangle$. Knowing that each worker has his own selfish threshold, and when the external condition reaches the threshold, the worker will have the power to engage in labor; we define the variable G_n^t to represent the intrinsic property of the worker. C_n^t means the objective cost of the behavior of participation in a task t for a node n. In addition, we have learned in the second section that the worker is not entirely rational and less likely to remain rational in every decision stage. Due to the cognition, experience, reference, and other psychological factors, the worker gets the conclusion that does not exactly match the objective fact when he analyzes and judges an external condition. So we define two functions A_n^t and AI_n^t. The former indicates the objective reward that a worker N can actually obtain when he participates in a perceived task T; the latter means his gains in his cognition. When a worker makes a decision, his actual reward in his cognition is the latter.

We define that each worker has two kinds of behaviors within the system: (a) choosing a sensing task and participating in perception and (b) sleeping. The main reason for preventing worker from participating in a sensing task is that there is the cost that must be paid in the process. Obviously, only when the worker thinks the reward he can get meets his own selfish threshold will he choose to participate in this perception task; otherwise he would rather sleep to prevent the loss of meaningless cost.

The worker maintains a real-time connection with the server in order to receive the task pushing at any time and chooses a satisfying task to perform according to his own conditions. The worker node sends its current decision information to the server platform, and the server platform updates the overall policy information in real time, then updates the participation of task, and gets the latest overall strategy and the information of task congestion.

3.3. The Construction of Loss Aversion in Crowdsensing

3.3.1. Basic Definitions. This paper needs to use the following concepts.

Definition 2 (the benefits of user nodes). The objective benefit of a user's involvement u_n^t in a sensing task is the difference between the reward he receives for his sensing task $P(t_n)$ and the cost of his participation in perception c_n^t, as follows:

$$u_n^t = P\left(t_n\right) - c_n^t, \quad \forall n \in N,\ t \in T. \tag{11}$$

Definition 3 (the congestion degree of a task). The sum of all nodes' contributions to a task is its congestion degree $\mathrm{Con}(t_i)$ as formula (12). The congestion degree represents the situation in which a task is currently executed by nodes.

$$\mathrm{Con}\left(t_i\right) = \sum_{n=1}^{n} \varphi_n^{t_i},$$
$$\text{where } \varphi_n^{t_i} = \begin{cases} 1, & s_n = t_i \\ 0, & s_n \neq t_i, \end{cases} \quad \forall t_i \in T,\ n \in N. \tag{12}$$

We artificially define $\varphi_n^{t_i}$ in formulas (12)–(15) as the contribution of a node to a task; its value is 0 or 1. Its value being 1 indicates that the node completes the task, and its value being 0 indicates that the node does not complete the task.

Definition 4 (social welfare). The social welfare U in crowdsensing is the sum of the benefits of all workers as follows:

$$U = \sum_{t_i \in T} \sum_{n \in N} \left(P\left(t_j\right) - c_n^{t_j}\right) \varphi_n^{t_j}. \tag{13}$$

Definition 5 (the average value of each data). The average value of each data $E(S)$ is the ratio of the total reward for all tasks in the system to the total number of data as follows:

$$E\left(S\right) = \frac{\sum_{t_i \in T} \sum_{n \in N} \left(P\left(t_j\right) - c_n^{t_j}\right) \varphi_n^{t_j}}{\sum_{n=1}^{n} \varphi_n^{t_i}}. \tag{14}$$

Definition 6 (cooperation rate). The rate of the total number of cooperators to the total number of nodes:

$$\text{Cooperation rate} = \frac{\sum^N \sum^T \varphi_n^{t_i}}{N}. \tag{15}$$

In order to measure the value of profit or loss from the reference point, and to successfully describe behavioral characteristics, the concrete expression of the value function is given as formula (16). This function explains three behavioral characteristics of the limited rational person: (a) most people are risk-averse when faced with the profit; (b) most people are risk-seeking when faced with the loss; (c) people are more sensitive to the loss than the profit.

$$v\left(\omega\right) = \begin{cases} \left(\omega - \omega_0\right)^{\alpha}, & \omega \gg \omega_0 \\ -\lambda\left(\omega_0 - \omega\right)^{\beta}, & \omega < \omega_0, \end{cases} \tag{16}$$

where ω_0 represents the reference point of the decision-maker, and if the gain is greater than the reference point, the decision-maker will perceive the profit, or else the loss will be perceived. α and β are the risk attitude coefficient, and λ is the loss aversion coefficient.

Table 2: Payoff matrix.

	Platform	
	Cooperation	Noncooperation
Worker		
Cooperation	$\left(-P\left(t_i\right), P\left(t_i\right) - c_n^t\right)$	$\left(0, -c_n^t\right)$
Noncooperation	$\left(-P\left(t_i\right), 0\right)$	$(0, 0)$

3.3.2. Payoff Matrix. This paper sets $P(t_i)$ as the objective reward that a node n participating in a task t_i can get, determined by the budget and the current congestion level of the task, as follows:

$$P\left(t_i\right) = \frac{B\left(t_i\right)}{\mathrm{Con}\left(t_i\right)} = \frac{B\left(t_i\right)}{\sum_{n=1}^{n} \varphi_n^{t_i}}. \tag{17}$$

The payoff matrix of nodes and platforms is shown in Table 2. Platform selects cooperation, that is, providing information and rewards of tasks for nodes. In the crowdsensing system, platform needs to do this all the time, which means platform keeps cooperation forever. The nodes select cooperation; that is, the nodes perform sensing tasks and feedback data as specified. This behavior needs to pay the cost, but also receive the appropriate reward; the nodes choose noncooperation, which are not involved in the perception of any task, keeping sleeping with paying nothing and receiving nothing.

3.4. The Reward Payment Algorithm Based on the Loss Aversion. The decision-making process of this paper focuses on the loss aversion in the decision-making of the nodes. In this way, we adjust the traditional pipelined payment model and divide the payment process into three stages: the release stage, the selection stage, and the settlement stage.

(a) The First Stage: The Release. In the release phase, first we establish the highest control authority for the server. It is reasonable. Although there is no centralized control to control the behavior of each individual as crowdsensing is a distributed system, the establishment of the common reputation mechanism and the virtual integration system are on the premise of the server's highest control authority.

As shown in Figure 3, in our system, for each node registered at the platform, the node obtains a system-specific part on its account, which allows the platform to pay or deduct reward to the node after joining in the crowdsensing system.

Before a sensing task is settled, this part of the node account that participates in the task is under the supervision of the server, and if the node exits the system in the middle, the part is recycled by the platform. Only when a perceived task is settled will the part of the reward be truly transferred to the node account and by its domination.

At this stage as described in Algorithm 1, the platform and the workers perform the following operations, respectively, as shown in Algorithm 1. The server platform collects sensing tasks from the requesters, generates task set, and extracts the properties of each task. The platform sorts and counts the

(1) **for** $n \in N$ **do**
(2) $n \leftarrow G_n^t, C_n^t$
(3) **if** n is available **do**
(4) Register on the platform
(5) Create a temporary account which is under the supervision of the system
(6) **else** quit the system
(7) **end if**
(8) $s_n = 0$
(9) **end for**
(10) **for** $t \in T$ **do**
(11) $t \leftarrow B(t_i), P(t_i), \mathrm{Con}(t_i)$
(12) **end for**

Algorithm 1: The release.

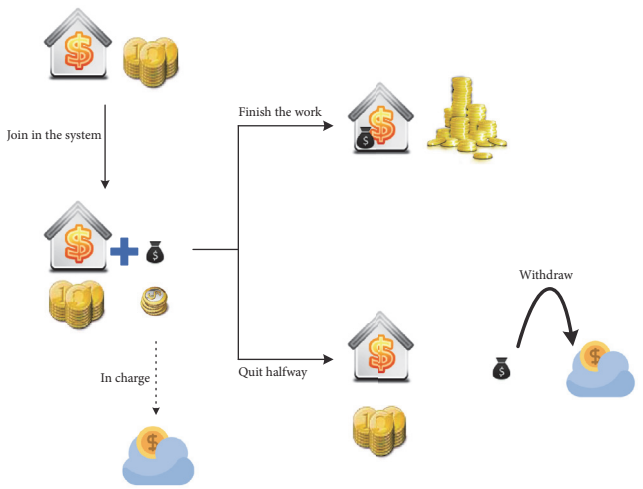


Figure 3: The crowdsensing-specific part on nodes' accounts.

registered users and generates the worker set. The platform pushes the task information to all users and gives payment vouchers and declares the rules to the node at the same time. The nodes that are able to perceive register on the platform and define their threshold attribute and cost attribute. The former indicates the demand of nodes, and the latter indicates the cost of nodes participating in task perception. The nodes need to keep the real-time communication with the platform to receive the push perception task of the platform. And the nodes understand the rules of the platform and are able to determine their own behavior.

(b) The Second Stage: The Assignment. In the second section, we already know that nodes are not infinitely greedy, and different nodes have different target thresholds because of their own factors. The selfish threshold of the node is the standard of how the node makes decisions.

(1) The Node Selects Noncooperation. From the common sense, it is easy to know that if the reward that a sensing task provided to a work node is not higher than the threshold of a selfish node, the node will not generate sufficient momentum to participate in perception. Its psychological state at this time is described as follows:

$$f(n;t) = \left\{\text{unsatisfied} \mid P(t_i) - c_n^t < G_n^t\right\}. \tag{18}$$

(2) The Node Selects Cooperation

(i) In the Traditional Incentive Mechanisms. When the node judges that the external conditions meet its own selfish threshold, the node will participate in perception. Its psychological state at this time is described as follows:

$$f(n;t) = \left\{\text{satisfied} \mid P(t_i) - c_n^t > G_n^t\right\}. \tag{19}$$

(ii) In the Loss Aversion Mechanism. The introduction of the loss aversion not only retains the rational characteristics, but also considers the situation of limited rational of nodes. Due to the loss aversion, the node makes not only the rational judgment, but also an additional judgment whether it can afford to lose the reward that platform puts in the system-specific part on its account. If the pain of the loss reaches a certain threshold, the node will choose to participate in the perception task to avoid this pain, thus contributing to cooperative behaviors. Its psychological state at this time is described as shown in the following:

$$f(n;t) = \left\{\text{satisfied} \mid \left\{\begin{array}{l} P(t_i) - c_n^t > G_n^t \\ \lambda P(t_i)^{\beta} > G_n^t \end{array}\right\}\right\}. \tag{20}$$

As described in Algorithm 2, in the selection phase of the node, the node selects the task that satisfies itself according to the above condition, and if it does not, it will sleep itself. At the same time as the node selection, the platform updates the congestion degree of tasks and the unit price of the compensation in real time. The node then judges whether to participate in the task according to the situation's change. Until all the nodes in the system can no longer find the task to maximize their own rewards, the stage stops.

(c) The Third Stage: The Settlement. Each task has a fixed opening time; platform settles the task when the time arrived.

(1) **When** there are users and tasks in the system **do**
(2) **for** every $n \in N$ **when** he is unsatisfied **do**
(3) Search every open task t_i
(4) Get the estimated price if participate in this task $P(t_i)$ from platform
(5) **if** $P(t_i) - c_n^t > G_n^t$ **do**
(6) Add t_i to List(node; time) = $\{t_j \mid t_j$ is satisfying$\}$
(7) **else if** $P(t_i) - c_n^t < G_n^t$ **do**
(8) **if** $\lambda P(t_i)^\beta > G_n^t$ **do**
(9) Add t_i to List(node; time) = $\{t_j \mid t_j$ is satisfying$\}$
(10) **end if**
(11) **end if**
(12) Choose the task $t^* = \arg\max P(t_i)$ and $t_i \in$ List(node; time)
(13) **end for**
(14) **Until** every user cannot find a valuable task

Algorithm 2: The assignment.

The platform checks whether the task has reached the settlement stage
(1) **for** $t_i \in T$ **do**
(2) Platform settles the task and issue a response command
(3) **for** every node $n \in$ userlist $= \{n \mid s_n = t_i\}$ **do**
(4) **if** it doesn't response in time **do**
(5) punish the node
(6) **else do**
(7) finish this assignment
(8) **end if**
(7) **end for**
(8) Platform process the data and feed back
(9) **end for**

Algorithm 3: The settlement.

The settlement phase as sketched in Algorithm 3 is mainly carried out as follows.

Platform

(i) Platform determines whether the task reached the settlement phase.

(ii) When the task reaches the settlement phase, the platform sends a settlement signal to the node receiving the task; the task that does not reach the settlement phase is not performed.

(iii) Platform makes the statistics of the sensed data feedback and thus finishes the transaction with the nodes that complete the task on time, and then reclaims rewards for the nodes that failed to finish on time.

(iv) The platform continues to push unfinished tasks.

(v) The platform will send the collected sensed data to the requester. This release ends.

Worker

(i) The nodes that decide to participate in the task do their work and send the sensed data back to the platform; the nodes that decide not to participate then give the rewards back.

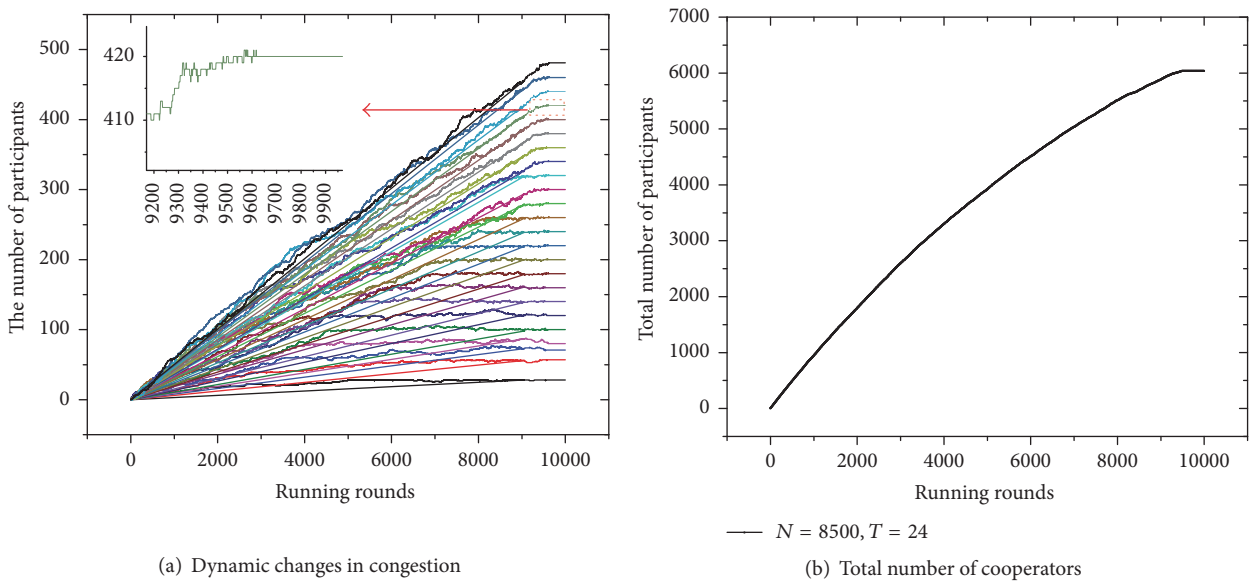
(ii) Nodes figure out the reward in the current round.

In the incentive mechanisms including virtual credit and reputation mechanisms, when we need not consider the security of payment in advance, the introduction of the loss aversion should be able to effectively improve the enthusiasm of the participants. In the monetary payment-type incentive, this paper sets up the account block structure in the release stage, so that we can effectively guarantee the security of unsettled tasks. We believe that because the monetary payment-type incentive is the most direct incentive, the effect of the loss aversion may be the most obvious.

The start of a task requires the requester to give the reward; those that want to get the data and refused to pay will not start a task at all, so the false-reporting behavior will not happen; and because of the settlement phase, those malicious nodes that want to take rewards but do not work return in vain, so put an end to the node's free-riding behavior.

4. Simulation

4.1. Parameters Setup. In order to evaluate the incentive effect of the loss aversion on nodes, we set up two simulation



(a) Dynamic changes in congestion

(b) Total number of cooperators

Figure 4: The dynamic change of the system selection process.

scenarios, including the loss aversion algorithm (LAA) and the completely rational algorithm (CRA) [49].

We set the CRA as a control group; in this set of simulations, the node is completely rational, and its judgment is mechanical. In the LAA, the node is limited rational and its loss aversion is easily aroused. We mainly analyze the algorithm performance from the cooperation rate and the average value of each data. The cooperation rate intuitively describes the incentive effect of the algorithm, while the average value of each dataset shows whether the system can acquire a sensed data at a lower average price. We evaluate the scalability of the system with the number of online nodes and the number of perceived tasks in the system. Evaluate the influence of nodes with different attributes on the system with nodes' gates and costs change. Evaluate the effect of the loss aversion level on the algorithm with the risk attitude coefficient and the loss aversion coefficient.

We assume that some tasks within the system require large amounts of data, so their reward budgets are high; some require small amount of data, so their reward budgets are low; for the simulation of this kind of situation, we make the reward budget of the sensing task t for $B_t = 40t$. The reward when a node n completes the perceived task t is R_n^t; its value is defined in Section 3.3.2; in this setting the task reward can get reasonable allocation, neither too high resulting in waste nor too low to attract workers. The node requirement G_n^t is a random number that belongs to $[1, G_{\max}]$. A node may have different requirements for each task or may be the same, which is determined by the node itself. C_n^t is the similar manner. In formula (16), parameters α and β are the risk attitude coefficients, and λ is the loss aversion coefficient. Their reference values are usually derived from the experimental results of the loss aversion presenters; that is, $\alpha = \beta = 0.88$ and $\lambda = 2.25$. We take these classical values as references to discuss the influence of the changes of these values on the results. The average of 50 times was taken in all experiments.

4.2. Analysis

4.2.1. Dynamic Changes of the Task Participation. Set $N = 8500$ and $T = 24$ to observe the dynamic changes of the system. Task selection is a dynamic process. The system constantly adjusts according to the different requirements of nodes in order to eventually find a sensing task to meet their requirements. In this process, due to the change of the congestion level of the task (rise up as a whole, because the nodes are constantly added, as shown in Figure 4(b)), the reward of payment to each node will also change. This may cause the situation that some nodes participate in the task when the congestion degree is low, but are not willing to participate when the congestion degree increases; then these nodes will exit the task to look for other tasks which meet their demands, resulting in the congestion degrees decreasing. Until finally all nodes find the satisfying tasks, the nodes with no satisfying tasks selected to sleep; the congestion curve of each task tends to be stable. Overall, the number of cooperators in the system is constantly increasing, which may have small fluctuations, because there are some nodes needing strategy adjustment, but in the end they can reach a stable situation.

4.2.2. The Number of Nodes and Number of Tasks. We compare the cooperation rates of the CRA and the LAA under different conditions to analyze the expansibility and stability of the system. We set $T = 24$ and compare the lower demand for nodes with $G_{\max} = 20$ and $C_{\max} = 10$ and the higher demand with $G_{\max} = 40$ and $C_{\max} = 20$. Figure 5(a) shows that increasing the online nodes can gain more data when the given sensing tasks are unchanged in the system. Still

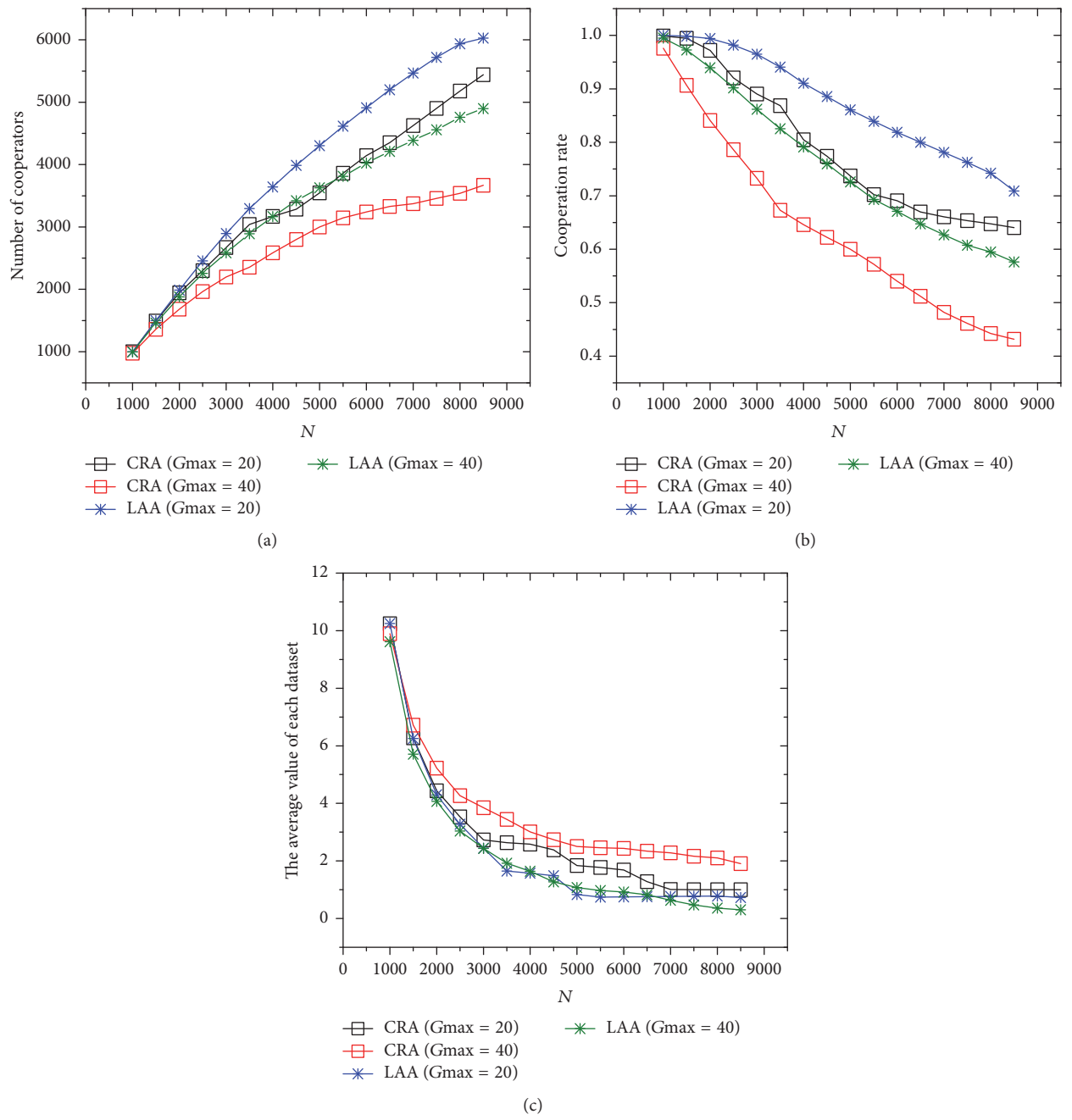


FIGURE 5: Impact of number of workers N in the LAA versus the CRA ($N = 5000$, $G_n^t \in [1, 20]$).

as shown in Figure 5(a), the superiority of the LAA is more pronounced when the congestion degree of the system is high. When $G_{\max} = 20$ and $N = 8500$, the cooperation rate of the CRA is 64%, while the cooperation rate of the LAA is 74.2%. When $G_{\max} = 40$ and $N = 8500$, the cooperation rate of the CRA was only 43%, while the LAA remains at 57%. This is due to the fact that the increase of the number of the nodes will cause the increase of the congestion degree of tasks. The available resources are fewer relative to the number of the nodes. The nodes in the LAA are more inclined to accept the tasks which are slightly lower than their own expectations because of their own psychological factors. This is more obvious in the case of nodes with higher demands. When there are fewer tasks meeting their own needs, the nodes in the LAA will make themselves willing to make some concessions under the strong psychological effect.

The average value of each dataset represents the average reward that the system needs to pay for an effective sensed data. The lower the value is, the more "cost-effective" the system is. Figure 5(b) shows the change in the average value of each dataset in the same situation. Since our tasks are budget-limited, their total budget is established. Therefore, when the number of participants is small, the reward assigned to each node will be very high; when the number of participants is

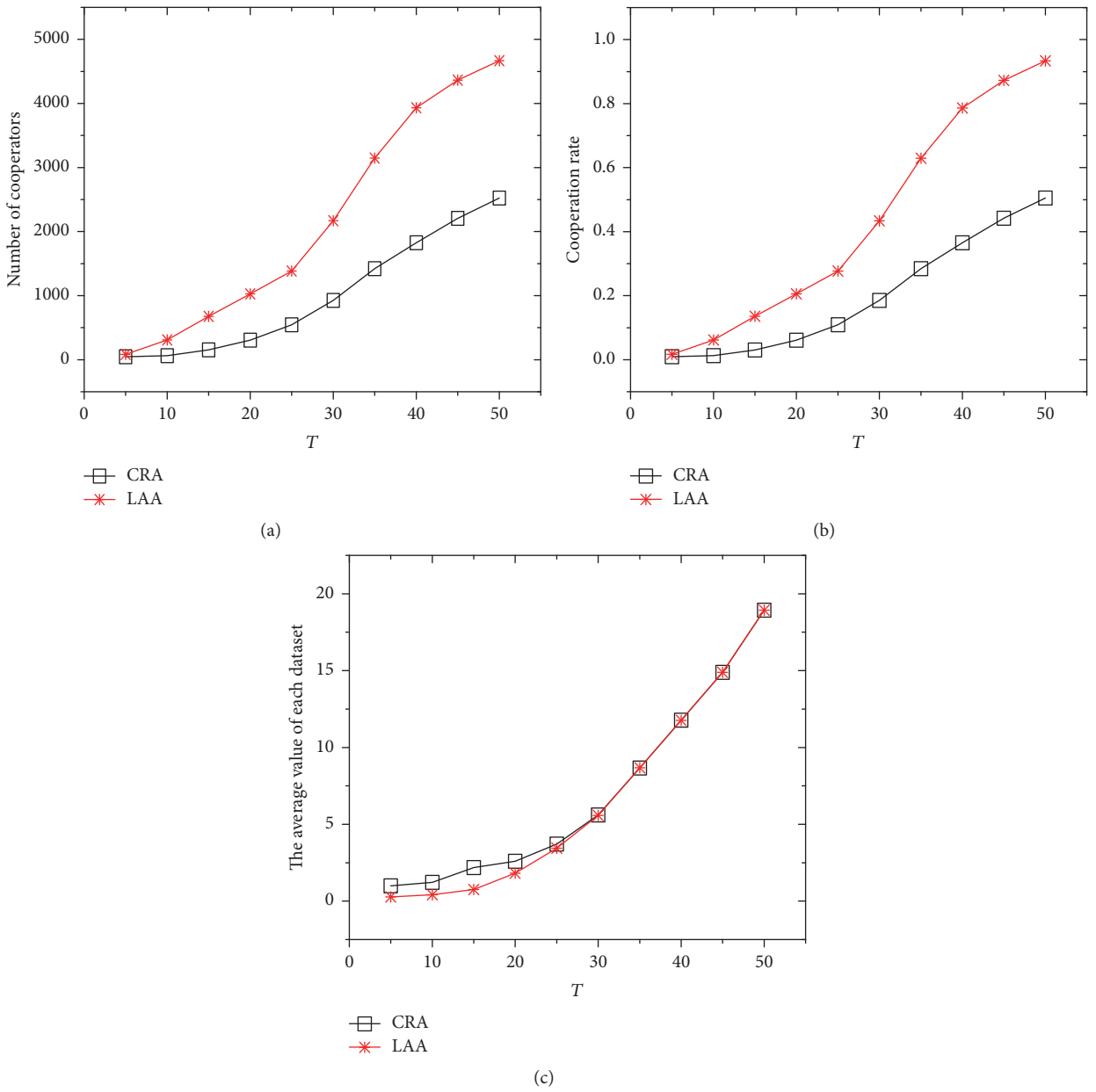


Figure 6: Impact of number of tasks T in the LAA versus the CRA ($N = 5000$, $G_n^t \in [1, 20]$).

too large, the reward is low. Too high unit reward causes waste, and too low reward cannot attract sufficient numbers of the nodes to work. When we fix the number of sensing tasks within the system and increase the number of the online nodes, at the beginning, the average value of each dataset is too high because the number of nodes is insufficient. In this case, the performance of the CRA and the LAA is similar, because nodes do not have to worry about the fact that they would find no suitable work when there is a surplus of resources. When the number of online nodes increases, the resources become insufficient. The effect of the LAA is very obvious; it can almost get the data at half the price of the CRA algorithm.

When the number of online nodes in the system does not change, we observe the changes in the number of cooperators with the increase in the number of sensing tasks in Figure 6. Obviously, when the number of tasks in the system is less, the number of participants will be less. However, when the system resources are extremely insufficient, we slowly add new resources (new sensing tasks) to the system and observe that the nodes in the LAA are more encouraged. This shows that, with the resource constraints, putting the same amount of new resources provokes more nodes in the LAA than the CRA. Similarly, when the number of online nodes is fixed and resources are changed, it can be seen that the LAA can keep a lower average value of each dataset when resources are extremely tight (when T is between 5 and 20).

4.2.3. Upper Bound of the Node Demand Threshold and the Cost. When the number of online nodes and the number

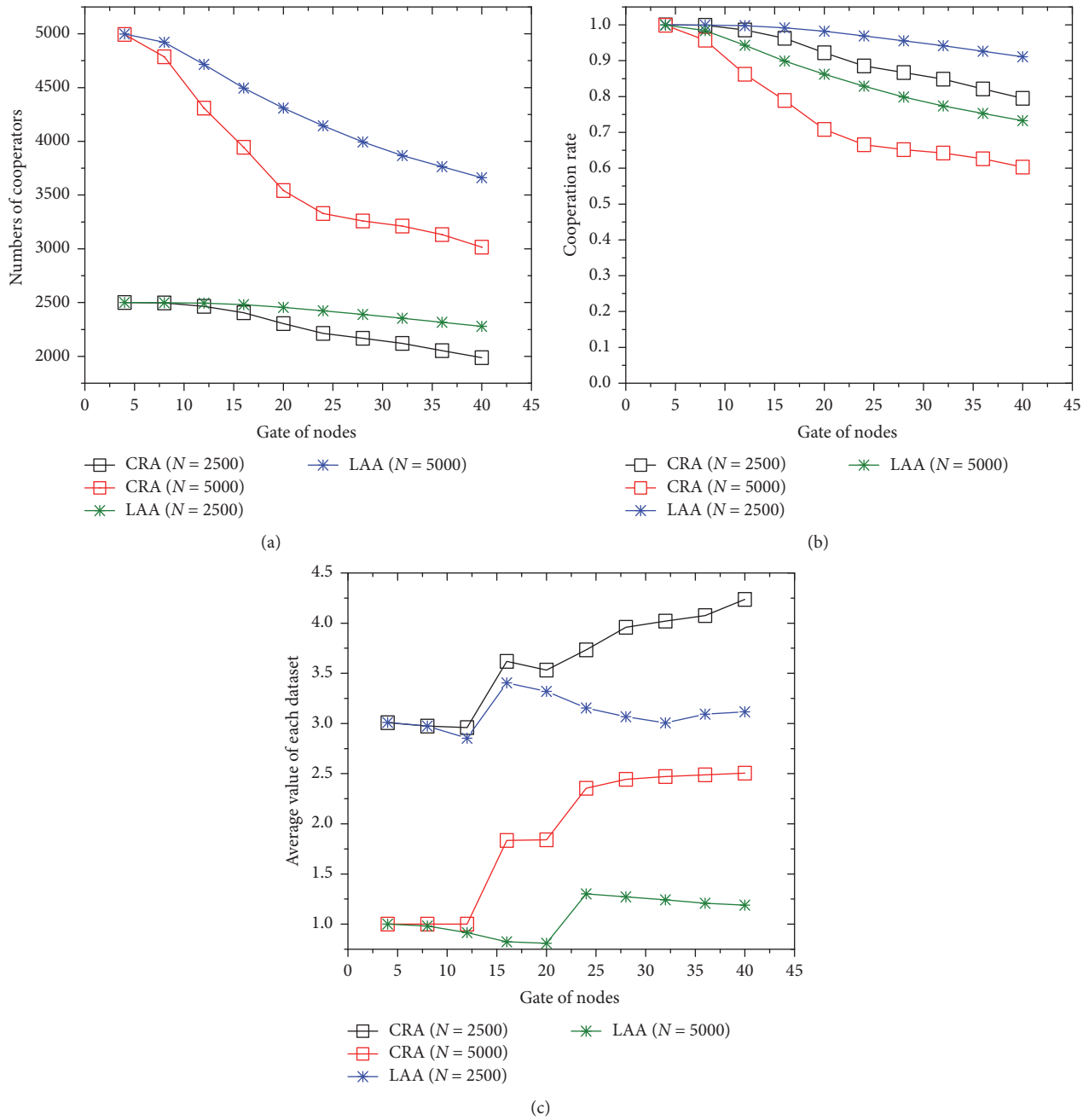


FIGURE 7: Impact of gate of nodes G_n^t in the LAA versus the CRA ($T = 24$, $N = 2500, 5000$).

of sensing tasks are unchanging, the selfish threshold of the node itself and the cost of its participation in the task also affect the cooperation rate. We compared the two cases where resources are more abundant ($N = 2500$, $T = 24$, $C_{\max} = 10$, and $C_n^t < G_n^t$) and resources are more insufficient ($N = 5000$,$T = 24$, $C_{\max} = 10$, and $C_n^t < G_n^t$). The nodes will participate in the tasks that meet their demands; this dynamic searching task process could give priority to low-demand nodes to find the right task, so as to ensure that the two algorithms will certainly make nodes cooperate. On one hand, the LAA's cooperation curve is significantly better than the CRA's. On the other hand, in Figure 7(a), comparing two curves when $N = 2500$ with those when $N = 5000$, we found when $G_{\max} = 40$, $N = 2500$, the CRA cooperation rate is 79.5%, while the LAA cooperation rate is 91.1%. In the case of $N = 5000$, the CRA cooperation rate is 60.3%, and the LAA cooperation rate is 74.3%. The difference between the CRA and the LAA is more obvious in the latter case. With the increase of $G_{\max}$, the decrease trend of the LAA curve slows down, which proves that the high demand node is more sensitive to the loss aversion.

The influence of the node threshold on the average value of each dataset is also obvious. Figure 7(c) compares them in case of $N = 2500$ and $N = 5000$, respectively. Since

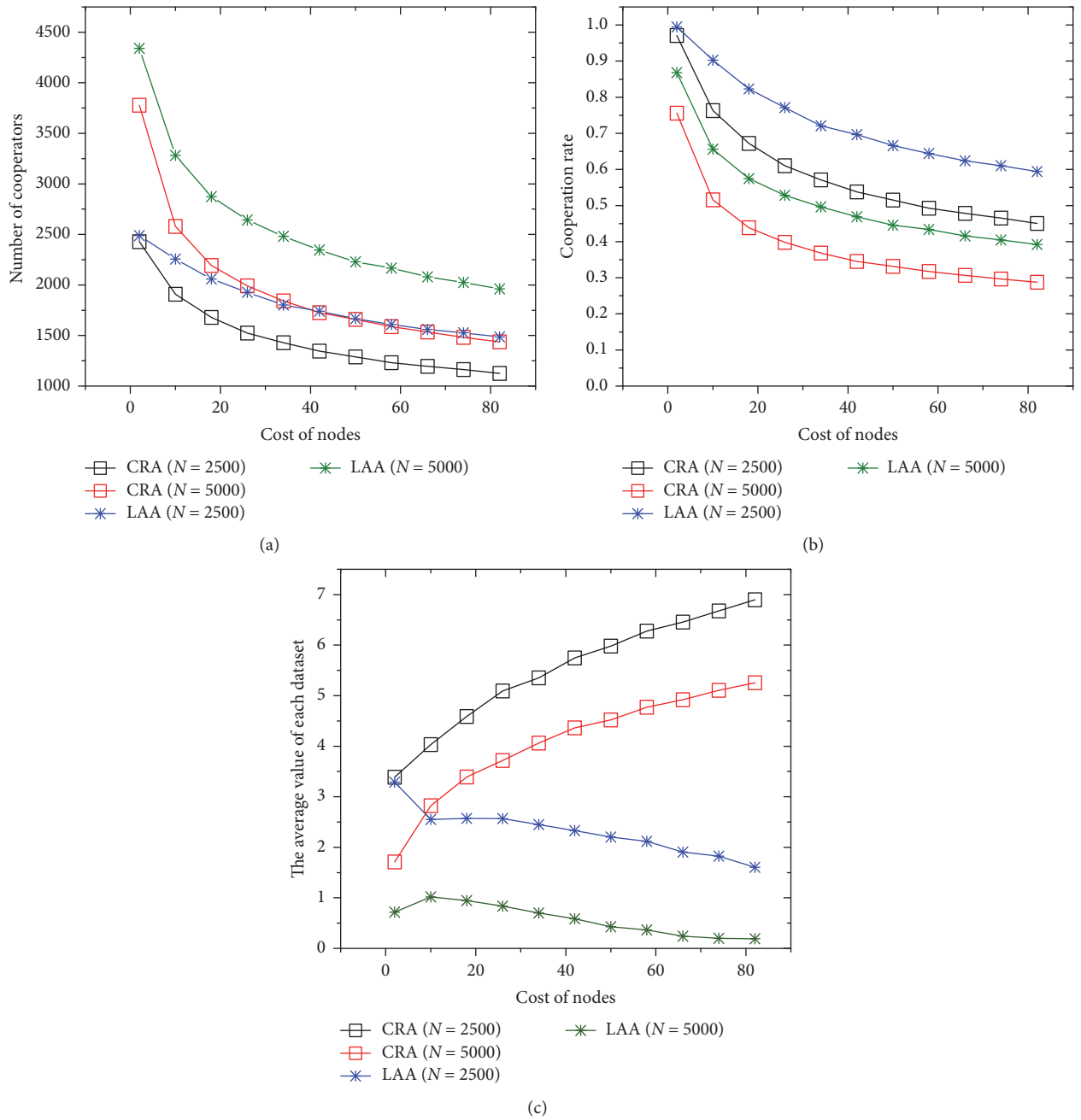


FIGURE 8: Impact of gate of nodes C_n^t in the LAA versus the CRA (T = 24, N = 2500, 5000).

the individuals and the number of participants in the system are changing (the costs of participating in the same task for different individuals are also different), it is normal that the data changed in a small range. So we only need to compare the differences between the two algorithms in the same case. In the case of $G_{\max}$ = 5, 10, the two datasets are similar. And when $G_{\max}$ >= 15, it indicates that the node's demand is increased. At this time, although the congestion degree does not change, the tasks which can satisfy the nodes are reduced. $G_{\max}$ increases; the average of each dataset becomes higher, which is reasonable; in this case the LAA can still obtain data at a relatively low price, reflecting the superiority of the LAA.

Besides the selfish threshold of the nodes, the cost of performing a task for nodes will also have a significant impact. In order to facilitate the analysis, we set $G_{\max}$ = 20, T = 24, and N = 2500, 5000 to compare the situations. C_n^t of the nodes is a random number that belongs to $[1, C_{\max}]$. Figures 8(a) and 8(b) show that as $C_{\max}$ increases, the cooperation rate decreases. When $C_{\max}$ = 80, N = 2500, the cooperation rate of the CRA is 45%, and the cooperation rate of the LAA is 59.4%. The latter is about 15% higher

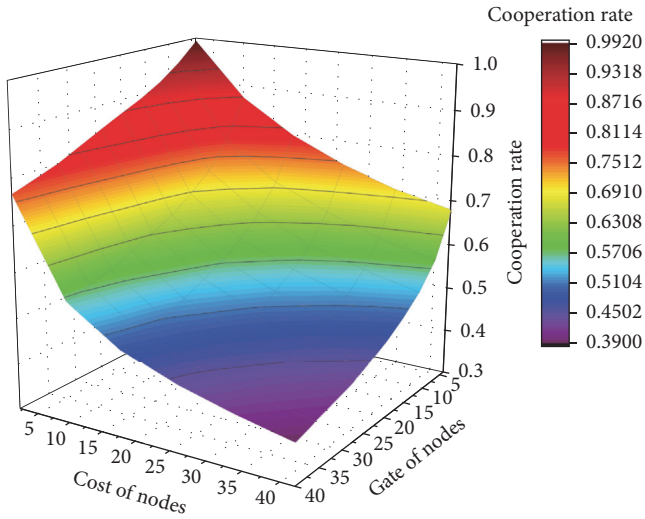


FIGURE 9: Impact of gate of nodes and cost of nodes versus CoopRate.

than the former. When $N = 5000$, the cooperation rate of the CRA is 28.7%, and the cooperation rate of the LAA is 39.8%.

Interestingly, we found that, with the growth of C_{max}, the average value of each dataset of the CRA is constantly higher, while the trend of the LAA is reduced in Figure 9. This is because, in the CRA algorithm, the main factor that a node considers is not the cost of the nodes, but the net reward (the reward minus the cost). As the cost increases, the number of tasks that can meet the needs of the nodes becomes less, the number of participating nodes becomes less, and the average value of each dataset increases. And for the LAA node, due to the loss aversion, in order to avoid the pain caused by the loss, the nodes accept the tasks as long as they did not want to lose the rewards that the task "has paid." So the number of partners within the system can be maintained at a high level; the average value of each dataset is reduced.

We compared the influences of C_{max} and G_{max} on the cooperation rate and still set the scale of $N = 5000$ and $T = 24$. It can be seen that the effect of C_{max} is larger than G_{max} in the LAA. For the same number of increments, C_{max} is more pronounced for the reduction of the cooperation rate. This is because the cost is objective; the LAA cannot reduce the values of the costs, but the human selfish threshold is relatively variable; the LAA reduces the node selfish threshold in fact by expanding its loss part. It is easier for nodes to compromise when resources become limited and then secondly to choose a suboptimal task.

4.2.4. The Risk Attitude Coefficient and the Loss Aversion Coefficient. For the risk attitude coefficient and the loss aversion coefficient, there are a number of discussions after the loss aversion has been proposed. In the experiments above this section, we all use $\beta = 0.88$ and $\lambda = 2.25$ as the node's loss aversion attribute. In this section, we discuss the changes about these two values, that is, how the degree of the loss aversion of a node will impact the algorithm. We set the analysis in the case of $N = 5000$, $T = 24$, $G_{max} = 60$, $C_{max} = 30$.

In order to compare with the completely rational algorithm, we take $\beta > 1$ and $\lambda = 1$ as the reference point (when the loss aversion coefficient is 1, it returns to the general model).

When $\beta > 0.8$ and $\lambda > 2.2$, the cooperation rate can still be maintained at a higher and more stable level as shown in Figure 10(a); the value is about 64%, and when $\beta = 0.3$ and $\lambda = 1.5$, the cooperation rate will be reduced by about 10% in the same case. For the average value of each dataset, it can be maintained below 1.6 when $\beta > 0.7$ and $\lambda > 2.0$, and when $\beta = 0.3$ and $\lambda = 1.5$, it needs to pay 2.4 units to get the data as shown in Figure 10(b).

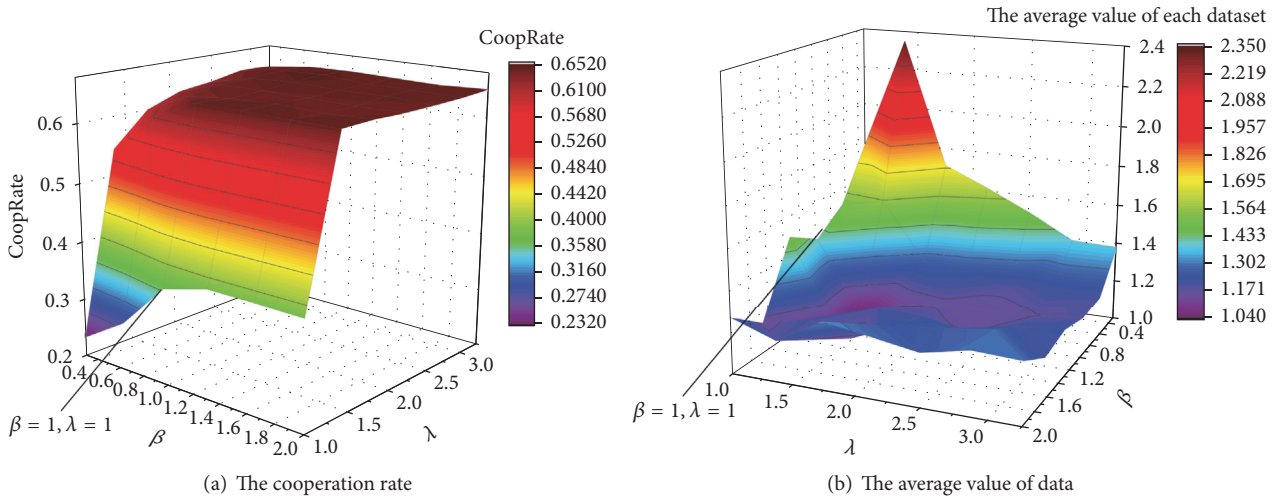
Considering the above two graphs, $\beta > 0.8$ and $\lambda > 2.2$ can maintain the higher cooperation rate of the system, but the average value of each dataset is also higher, which is not cost-effective for the system. And when $\beta > 0.7$ and $\lambda > 2.0$, although the average value of each dataset is low, the cooperation rate is not high, which might not collect enough data. Only when the loss of a node is in the two ranges, that is, $0.8 < \beta < 2.0$ and $2.0 < \lambda < 3.5$, can the system collect enough data and purchase data at a lower price, which can guarantee the quality and prices of the sensed data at the same time.

5. Conclusion

Cooperative guarantee is always a hotspot in the crowdsensing system. For greedy nodes, the benefits are, of course, the higher the better, but the limited budget makes the resources in crowdsensing in most cases insufficient. This is the basic contradiction in the crowdsensing. In order to alleviate this contradiction, the traditional incentive mechanisms under the premise of rational individuals designed a lot of external mechanisms to regulate the behaviors of nodes. These incentive mechanisms promote the cooperative behaviors of the nodes from a variety of angles, but how to design a reasonable internal mechanism for the cooperation is still an unresolved problem.

This paper presents an incentive mechanism LAA based on the limited rational premise, which makes full use of the psychological activities that people cannot ignore in the decision-making process. It emphasizes the sensitive characteristics of the node to the loss, which makes the node expand the value of the lost parts irrationally in its cognition. By adjusting the architecture and the process of the payment algorithm, we have stimulated the loss aversion of the nodes, making the nodes more active in the sensing task. Finally, we use the experimental data to analyze the efficiency of the algorithm.

We have realized that considering irrational factors to solve the problem may have a multiplier effect, and the nature of the cooperation is still to be discussed more; in the next step, we consider the following: (1) comparing the loss aversion algorithm with the more current incentive mechanism to further improve the performance of crowdsensing systems; (2) considering more irrational factors and stimulating the cooperative psychology of the node from the



(a) The cooperation rate

(b) The average value of data

FIGURE 10: Impact of β and λ versus CoopRate and average value of each data.

framing bias, endowment effect, choice architecture, and so on. We believe that irrationality thinking can open up new ideas for crowdsensing systems.

Parameters

N: The number of online nodes in the system, [1000, 8500]
T: The number of sensing tasks in the system, [5, 50]
$B(t_i)$: The reward budget for each perceived task; the value is $40 * i$
$P(t_i)$: A reward scheme for the payment of a perceived task to a single participant aware node; the value is
G_{max}: Upper bound of node demand threshold; we set it 20, 40, 60
C_{max}: The node cost, related to G_{max}; the value is $u * G_{max}$
u: A coefficient of the cost to the demand; the value is 0.5, 1, 2
α, β: The risk attitude coefficient; the value is [0.3, 2]
λ: The loss aversion coefficient; the value is [1, 3.25].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant 61572528.

References

[1] F. Restuccia, S. K. Das, and J. Payton, "Incentive mechanisms for participatory sensing: Survey and research challenges," *ACM Transactions on Sensor Networks*, vol. 12, no. 2, article no. 13, 2016.

[2] J. Dutta, F. Gazi, S. Roy, and C. Chowdhury, "AirSense: Opportunistic crowd-sensing based air quality monitoring system for smart city," in *Proceedings of the 15th IEEE Sensors Conference, SENSORS '16*, USA, 2016.

[3] J. Wan, J. Liu, Z. Shao, A. V. Vasilakos, M. Imran, and K. Zhou, "Mobile crowd sensing for traffic prediction in internet of vehicles," *Sensors*, vol. 16, no. 1, article 88, 2016.

[4] T. Xing, B. Xie, T. Xian et al., "Treasures status monitoring based on dynamic link-sensing," *Peer-to-Peer Networking and Applications*, vol. 10, no. 3, pp. 780–794, 2017.

[5] B. Langguth, A. B. Elgoyhen, and W. Schlee, "Potassium channels as promising new targets for pharmacologic treatment of tinnitus: Can Internet-based crowd sensing initiated by patients speed up the transition from bench to bedside?" *Expert Opinion on Therapeutic Targets*, vol. 20, no. 3, pp. 251–254, 2016.

[6] Y. Xiong, D. Shi X, B. Ding et al., "Survey of Mobile Sensing," *Computer Science*, 2014.

[7] R. Schmid, K. Schneeberger, and M. Taborsky, "Feel good, do good? Disentangling reciprocity from unconditional prosociality," *Ethology*, vol. 123, no. 9, pp. 640–647, 2017.

[8] A. P. Fiske, "The four elementary forms of sociality: framework for a unified theory of social relations," *Psychological Review*, vol. 99, no. 4, pp. 689–723, 1992.

[9] M. Riahi, R. Rahman, and K. Aberer, "Privacy, Trust and Incentives in Participatory Sensing," in *Participatory Sensing, Opinions and Collective Awareness*, Understanding Complex Systems, pp. 93–114, Springer International Publishing, Cham, 2017.

[10] D. Zhao, X.-Y. Li, and H. Ma, "Budget-feasible online incentive mechanisms for crowdsourcing tasks truthfully," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 647–661, 2016.

[11] Y. Fan, H. Sun, and X. Liu, "Truthful incentive mechanisms for dynamic and heterogeneous tasks in mobile crowdsourcing," in *Proceedings of the 27th IEEE International Conference on Tools with Artificial Intelligence, ICTAI '15*, pp. 881–888, Italy, November 2015.

[12] J. Mukhopadhyay, A. Pal, S. Mukhopadhyay et al., "Quality adaptive online double auction in participatory sensing," 2017.

[13] B. Song, H. Shah-Mansouri, and V. W. S. Wong, "Quality of Sensing Aware Budget Feasible Mechanism for Mobile Crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3619–3631, 2017.

[14] Y. Zhang, H. Zhang, S. Tang, and S. Zhong, "Designing Secure and Dependable Mobile Sensing Mechanisms with Revenue Guarantees," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 100–113, 2016.

[15] T. Luo, S. K. Das, H. P. Tan, and L. Xia, "Incentive mechanism design for crowdsourcing: An all-pay auction approach," *ACM Transactions on Intelligent Systems and Technology*, vol. 7, no. 3, article no. 35, 2016.

[16] J. Mukhopadhyay, V. Singh K, S. Mukhopadhyay et al., "Online Participatory Sensing in Double Auction Environment with Location Information," 2017.

[17] L. Gao, F. Hou, and J. Huang, "Providing long-term participation incentive in participatory sensing," in *Proceedings of the 34th IEEE Annual Conference on Computer Communications and Networks (IEEE INFOCOM '15)*, pp. 2803–2811, Hong Kong, China, May 2015.

[18] X. Zhang, Z. Yang, W. Sun et al., "Incentives for mobile crowd sensing: a survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 54–67, 2016.

[19] H. Xie, J. C. S. Lui, and D. Towsley, "Incentive and reputation mechanisms for online crowdsourcing systems," in *Proceedings of the 23rd IEEE International Symposium on Quality of Service, IWQoS '15*, pp. 207–212, USA, 2015.

[20] F. A. Santos, T. H. Silva, T. Braun, A. A. F. Loureiro, and L. A. Villas, "Towards a sustainable people-centric sensing," in *Proceedings of the IEEE International Conference on Communications, ICC '17*, France, May 2017.

[21] L. Jiang, F. He, Y. Wang, L. Sun, and H. Huang, "Quality-Aware Incentive Mechanism for Mobile Crowd Sensing," *Journal of Sensors*, vol. 2017, no. 3, pp. 1–14, 2017.

[22] K. Richter, "Identifying Landmark Candidates Beyond Toy Examples," *KI - Künstliche Intelligenz*, vol. 31, no. 2, pp. 135–139, 2017.

[23] J. Liu, H. Shen, and X. Zhang, "A survey of mobile crowdsensing techniques: A critical component for the internet of things," in *Proceedings of the 25th International Conference on Computer Communications and Networks, ICCCN '16*, USA, August 2016.

[24] T. Luo and L. Zeynalvand, "Reshaping Mobile Crowd Sensing Using Cross Validation to Improve Data Credibility," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '17)*, pp. 1–7, Singapore, December 2017.

[25] R.-I. Ciobanu, R.-C. Marin, C. Dobre, and V. Cristea, "ONSIDESELF: A Selfish Node Detection and Incentive Mechanism for Opportunistic Dissemination," in *Internet of Things (IoT) in 5G Mobile Technologies*, Springer International Publishing, 2016.

[26] T. Luo, S. S. Kanhere, J. Huang, S. K. Das, and F. Wu, "Sustainable incentives for mobile crowdsensing: Auctions, lotteries, and trust and reputation systems," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 68–74, 2017.

[27] S. Noor, R. Hasan, and A. Arora, "ParkBid: An Incentive Based Crowdsourced Bidding Service for Parking Reservation," in *Proceedings of the IEEE International Conference on Services Computing (SCC '17)*, pp. 60–67, Honolulu, HI, USA, June 2017.

[28] X. Duan, C. Zhao, S. He, P. Cheng, and J. Zhang, "Distributed algorithms to compute walrasian equilibrium in mobile crowdsensing," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4048–4057, 2017.

[29] H. Gao, C. H. Liu, W. Wang et al., "A survey of incentive mechanisms for participatory sensing," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 918–943, 2015.

[30] Y. Wu, J. R. Zeng, H. Peng, H. Chen, and C. P. Li, "Survey on incentive mechanisms for crowd sensing," *Journal of Software. Ruanjian Xuebao*, vol. 27, no. 8, pp. 2025–2047, 2016.

[31] D. Muller, "The Anatomy of Distributional Preferences with Group Identity - Early draft, please do not circulate," *Working Papers*, 2017.

[32] K. Chang, M. N. Young, M. I. Hildawa et al., "Portfolio Selection Problem Considering Behavioral Stocks," in *Proceedings of the World Congress on Engineering, The International Conference of Financial Engineering*, 2015.

[33] F. R. Tobias, "To consume or to save: are we maximizing or what?" *Handbook of Behavioural Economics and Smart Decision-Making*, 2017.

[34] "Theories of Economic Decision-Making: Value, Risk and Affect," in *Economic Psychology*, John Wiley & Sons, Ltd, chapter 2 edition, 2017.

[35] P. Lacour, "Experimenting with Social Norms: Fairness and Punishment in Cross-Cultural Perspective," *Eastern Economic Journal*, vol. 43, no. 2, pp. 372–374, 2017.

[36] D. Kahneman, *Schnelles Denken, langsames Denken*, Siedler Verlag, 2015.

[37] A. Tversky and D. Kahneman, "Advances in prospect theory: cumulative representation of uncertainty," *Journal of Risk and Uncertainty*, vol. 5, no. 4, pp. 297–323, 1992.

[38] X. Li, "Allais Paradox revisit: The representing frame in the decision with three possible outcomes using the equate-to-differentiate model: Better-worse or best-worst comparison," *Acta Psychologica Sinica*, vol. 49, no. 2, p. 262, 2017.

[39] P. Mandal, R. Kaul, and T. Jain, "Stocking and pricing decisions under endogenous demand and reference point effects," *European Journal of Operational Research*, vol. 264, no. 1, pp. 181–199, 2018.

[40] A. Howes, P. A. Warren, G. Farmer, W. El-Deredy, and R. L. Lewis, "Why contextual preference reversals maximize expected value," *Psychological Review*, vol. 123, no. 4, pp. 368–391, 2016.

[41] A. Scott and J. Witt, "Loss Aversion, Reference Dependence and Diminishing Sensitivity in Choice Experiments," *Social Science Electronic Publishing*, 2015.

[42] A. Bilbao-Terol, M. Arenas-Parra, V. Cañal-Fernández, and C. Bilbao-Terol, "Multi-criteria decision making for choosing socially responsible investment within a behavioral portfolio theory framework: a new way of investing into a crisis environment," *Annals of Operations Research*, vol. 247, no. 2, pp. 549–580, 2016.

[43] S. Bhatia, "Comparing theories of reference-dependent choice," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 43, no. 9, pp. 1490–1507, 2017.

[44] J. Yu, M. H. Cheung, J. Huang, and H. V. Poor, "Mobile Data Trading: Behavioral Economics Analysis and Algorithm Design," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 4, pp. 994–1005, 2017.

[45] j. Ruß and S. Schelling, "Multi Cumulative Prospect Theory and the Demand for Cliquet-Style Guarantees," *Journal of Risk & Insurance*, 2017.

[46] I. Iturbe-Ormaetxe, G. Ponti, and J. Tomás, "Myopic loss aversion under ambiguity and gender effects," *PLoS ONE*, vol. 11, no. 12, Article ID e0161477, 2016.

[47] M. Schlüter, A. Baeza, G. Dressler et al., "A framework for mapping and comparing behavioural theories in models of social-ecological systems," *Ecological Economics*, vol. 131, pp. 21–35, 2017.

[48] S. J. Dubner and S. D. Levitt, "Monkey business: can capuchins understand money," in *The New York Times Magazine*, 2005.

[49] Y. Sun, Y. Zhu, Z. Feng, and J. Yu, "Sensing processes participation game of smartphones in participatory sensing systems," in *Proceedings of the 11th Annual IEEE International Conference on Sensing, Communication, and Networking, SECON 2014*, pp. 239–247, Singapore, July 2014.

Hindawi
Scientific Programming
Volume 2018, Article ID 9308742, 10 pages
https://doi.org/10.1155/2018/9308742

Research Article

Research on Monitoring and Prewarning System of Accident in the Coal Mine Based on Big Data

Xu Xia,[1,2] **Zhigang Chen**,[1] **and Wei Wei**[3]

[1]*School of Software, Central South University, Changsha, China*
[2]*Hunan Vocational Institute of Safety Technology, Changsha, China*
[3]*School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China*

Correspondence should be addressed to Zhigang Chen; czg@csu.edu.cn

Received 25 October 2017; Accepted 5 December 2017; Published 6 March 2018

Academic Editor: Wenbing Zhao

More and more big data come from sensor nodes. There are many sensor nodes placed in the monitoring and prewarning system of the coal mine in China for the purpose of monitoring the state of the environment. It works every day and forms the coal mine big data. Traditional coal mine monitoring and prewarning systems are mainly based on mine communication cable, but they are difficult to place at coal working face tunnels. We use WSN to replace mine communication cable and build the monitoring and prewarning system. The sensor nodes in WSN are energy limited and the sensor data are complicated so it is very difficult to use these data directly to prewarn the accident. To solve these problems, in this paper, a new data aggregation strategy and fuzzy comprehensive assessment model are proposed. Simulations compared the energy consumption, delay time, cooperation cost, and prewarning time with our previous work. The result shows our method is reasonable.

1. Introduction

Coal is the main energy source in China. According to 2016 National Economic and Social Development Statistical Bulletin, the coal consumption in 2015 is about 64% of total energy consumption. So the government of China is paying more and more attention to safety production and proposing to use WSN, big data, IOT, and AI technologies to build "digital coal mine" so as to improve the safety level of coal mine industry. In China, most of the coal mining has happened underground, so it is very complicated about the environment in the coal mine. Some gases such as CH4 and CO are easy to gather in the coal mine tunnels; it is the main reason causing explosion accident; many workers lose their life. Since 2010, all of the coal mine industries were asked to install the monitoring system to prevent the happening of the accident. The monitoring system is working 24 hours a day without interruption. It means there are lots of monitoring data produced in the monitoring system every day. These data mainly include the state of equipment, the concertation of gas, the pressure of roof, the speed of the wind, and so on. These data have the characteristics of big data: large data, many types, high velocity, high value, and complex processing process [1]. Taking the data obtained by monitoring system by the State Administration of Work Safety in 2015 as an example, the cumulative information exceeds 5 million and the space occupied is 10TB [2]. We can use these data to build the suitable prewarning model and decrease the accident. It is meaningful to society.

The traditional communication method of the monitoring system is burying mine communication cable underground, but it is difficult to do at some places such as coal working face because the coal working face always changes with the digging process. With the development of WSN, its characteristics were proved to easily be used in industry [3–5]. For these reasons, many scholars proposed using WSN to replace the mine communication cable of the monitoring system. Obviously, the features of coal mine industry are much different with other fields when we use WSN; the coal mine tunnels are very long and narrow. For example, in the mines of the Nanyang Coal Industry Co., Ltd., Hengyang, China, the main haulage roadway is approximately 12,000 m

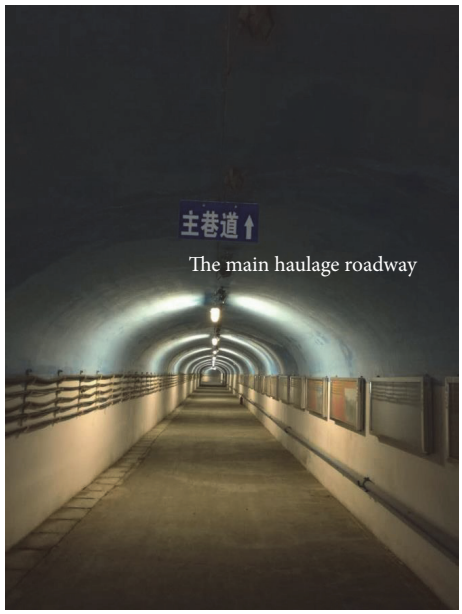


Figure 1: The main haulage roadway.

long, and most return airways have lengths of more than 1000 m, but the width is only several meters (Figure 1 shows part of the main haulage roadway). If we want to use WSN in the monitoring system in the coal mine industry, we should solve some problems. The big problem is how to extend the lifetime because it is difficult to change the battery of sensor.

The monitoring system of coal mine almost works every day. So it can produce big data. If we use these data reasonably, we can know the accident in advance and take some measurement to prevent the happening of the accident. Reference [6] proposed if we want to use big data in the coal mine industry with WSN, we should consider how to decrease the energy consumption. Reference [7] proposed that data aggregation can be used in WSN and saved more energy. Based on above research work, this paper will focus on how to use WSN in the monitoring and prewarning system based on big data in coal mine industry.

This paper is organized as follows: Section 2 describes related studies. Section 3 explains the design of the monitoring and prewarning system. Section 4 presents the data aggregation strategy and prewarning model. Section 5 presents the simulation and analyzes the performance. The final section provides the conclusion and future research directions.

2. Related Studies

Big data is not a new concept now. But earlier, the big data was limited to some specific organizations like Google, Microsoft, Yahoo, and so on. However, with the developments of IOT, cloud computing and sensor network, the cost of hardware is decreasing and the storage and processing power is increasing. As a result, many sources such as sensors and applications start to generate more and more data. The organizations tend to store these data easily for a long time because the storage and processing ability are great.

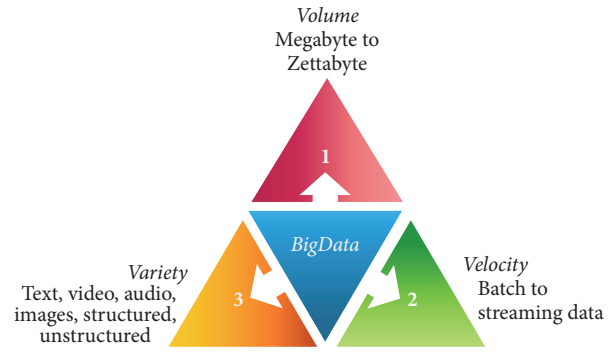


Figure 2: Characteristics of big data.

In fact, there is no uniform definition for "big data." The well-known definition is 3V's [8]: volume, variety, and velocity (as shown in Figure 2). EMC [9] has defined big data as any attribute that challenges constraints of a system capability or business need.

More and more big data come from sensor nodes. Reference [10] proposed a greenhouse gas sensor network located throughout California where it collects a large number of real-time data about greenhouse gases and their behavior. The project [11] embedded about 200 sensor nodes on the bridge to monitor the state of the bridge. This monitoring system collects a variety of data including temperature and the pressure of the bridge's concrete reaction to any change. Sensor nodes can collect information in the natural disaster situation in order to optimally utilize the resource and manage supply chains [12]. The challenges in big data mainly include two categories: engineering and semantic [13]. The Jet Propulsion Laboratory (JPL) has identified a number of major challenges in big data [14]; it includes the energy problem especially for the sensor nodes because there are more and more big data coming from sensor nodes in the future.

As to the problem of energy, data aggregation is an efficient method to decrease energy consumption and prolong the lifetime of WSN [15]. To reduce the amount of communication data in WSN, a lot of correlation-based data aggregation methods have been proposed in [7, 16–24]. The traditional data aggregation methods which are used in WSN mainly include two categories, the first type is based on least square method, Bayesian estimation method, D-S evidence theory, and so on; the other is based on artificial intelligence theory of artificial neural network method, fuzzy reasoning method, and rough set method [25]. Reference [26] introduced the data density correlation degree to decrease the amount of data conveyed to sink node, so it can help save more energy.

However, most of the previous studies do not focus on the coal mine. Only a few studies are about coal mining. References [3, 27–30] have studied the energy consumption problem of the coal mine, but all of them did not consider the big data technology and only use single-sink structure. Some novel algorithm is proposed in [31, 32]: in [31] a model for underground mines is generated by adopting a performance-based approach; in [32] a green MAC algorithm is proposed for smart home sensor networks. These strategies

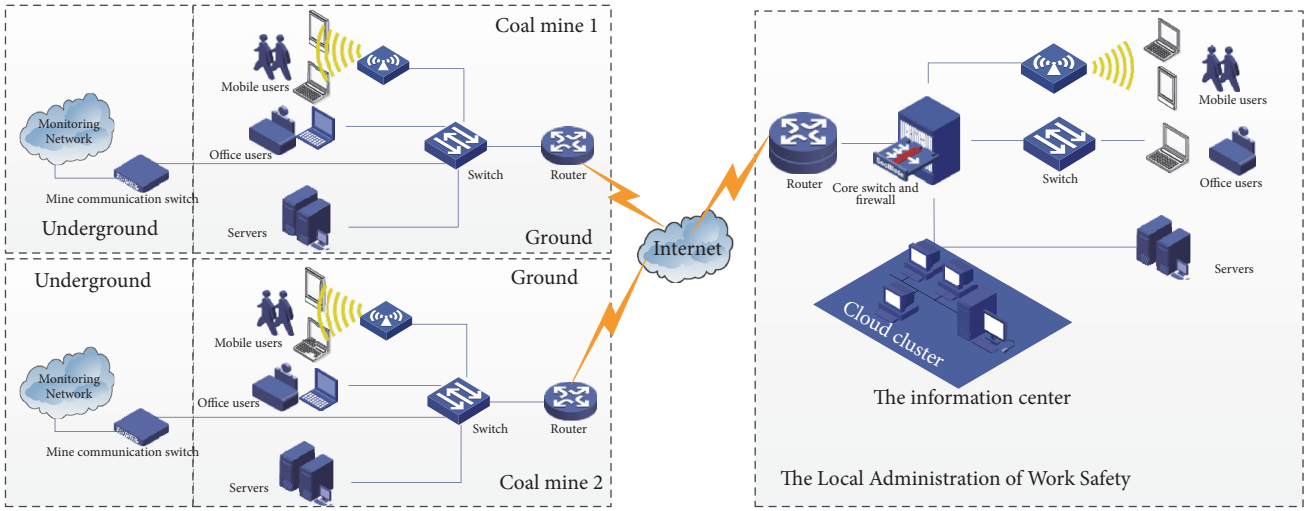


Figure 3: The overall structure of system.

effectively extend the network lifetime and improve network performance. Reference [33] introduced IOT and WSN in a mine and described the mine production, monitoring, and prewarning system based on big data technology, but it did not explain how to use sensor nodes and how to build the prewarning model. References [34–44] designed an architecture of monitoring system based on WSN and discussed a data aggregation strategy but did not consider using big data.

Our paper focuses on how to design the monitoring and prewarning system suitable for coal mine industry, how to use data aggregation strategy to decrease energy consumption, and how to build a prewarning model to prevent the accident happening in advance.

3. Design of the Monitoring and Prewarning System

3.1. Overall Structure of System. The Local Administration of Work Safety requires knowing the real-time production state of each coal mine, so all of the monitoring data are sent to the information center of the Local Administration of Work Safety. We design the structure of monitoring and prewarning system based on big data which is shown in Figure 3.

The monitoring data are sent to the cloud cluster and form big data; then we can use furry comprehensive assessment model (explained in Section 4.2) to prewarn the problems of the coal mine in advance. The officers and workers in Local Administration of Work Safety and coal mines can gain the prewarning message through their mobile phone or computer, and then they can make some decisions and measures to solve the problems to avoid the accident in advance [23, 30].

3.2. Structure of Monitoring Network. The monitoring network is located in underground. There are a large number of monitoring data, mainly the data generated by sensors in the WSN, that are generated in the production process, like the values of voltages, the concentration of gas, the speed of the wind, the pressure of roof, and so on [33]. In our study of the monitoring system, we focus on the monitoring of the concentration of CH4; it is the main reason for explosion and fire accident in the coal mine. We can use the same method to monitor and process other gases such as CO.

The Coalmining tunnels in China usually are categorized with their functions, including development tunnels, preparation tunnels, and mining tunnels [23, 30, 35–44]. The development tunnels are served for the whole mine, including the horizontal mining area, such as the main haulage roadways and the main return airways. The preparation tunnels are used for digging tunnels such as upward and downward mining areas. The mining tunnels are used to form the coal working face, such as the return airflow roadway and the haulage roadways of coal working face.

The main haulage roadways are relatively wide conveniently to bury mine communication cable. Along the development tunnels, there are many branches; most of them are coal working face tunnels. The coal working face tunnels are narrow, irregular, and always changing with the development of coal mining, so it is difficult to bury mine communication cables; we use WSN to replace the mine communication cable. For these considerations, in our study, the structure of monitoring network is shown in Figure 4.

The WSN is composed of two types of nodes: sink nodes and sensor nodes. The sink nodes are placed near the junction of haulage roadways and coal working face tunnels; it means the number of sink nodes is equal to the number of branches. The sink nodes connect to each other through the mine communication cable. The sensor nodes are placed at the top and the middle of the roof, because the CH4 is lighter than air and always gather together at the top of the roof. The sensor nodes in each coal working face tunnel have their own ID; the ID number increases along with coal working face tunnel; it means the ID of the first sensor node near the sink node is 1; then its neighbor's ID is 2. In order to gain stable communication performance, we use the Mine Segmenting Wireless Channel Model [31]. The sensor nodes can

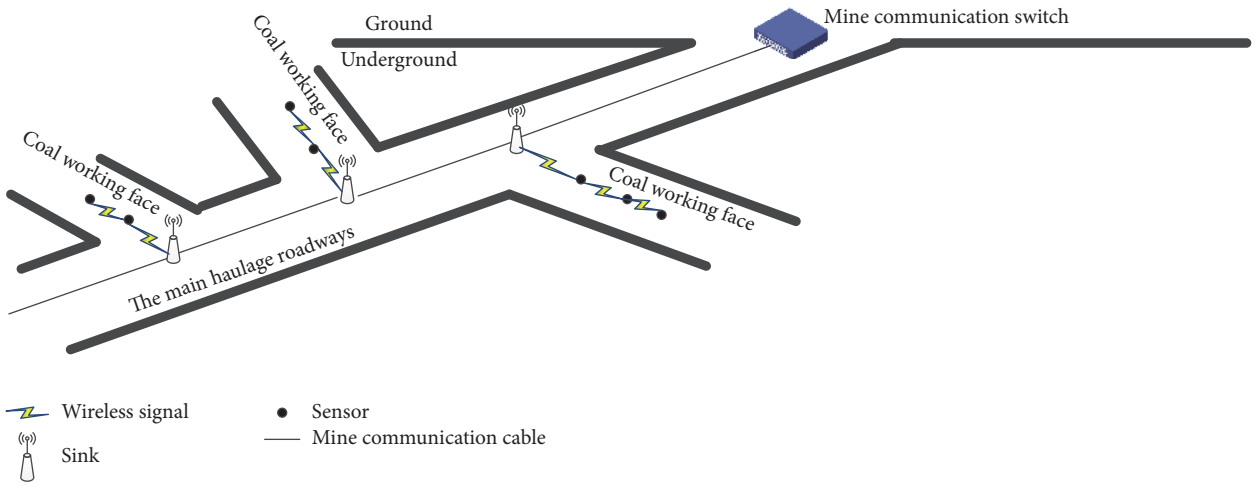


Figure 4: The structure of monitoring network.

communicate with neighbor sensor nodes within two hops. The sensor nodes send data to the sink nodes in one or two hops.

In this paper, we suppose that the number of sensor nodes in the network is N, and they are distributed in a long-strip region measuring $L \times W$ with $L \gg W$. As soon as the sink nodes and sensor nodes are deployed, the location is fixed and it no longer changes; the output power of sensor nodes is adjustable according to [31]. The sensor nodes are isomorphic with the same initial energy, and they also have data fusion function and self-sensing of residual energy. Furthermore, the energy of sink nodes is unlimited.

3.3. Working Mode. The monitoring network is always in working state for 24 hours without interruption, so the sensor nodes can collect and transfer data permanently; it means the sensor nodes will consume much energy. As we all know that the sensor nodes are powered by the battery which energy is limited and the consumption of energy mainly comes from the communication process. So in our study, we propose the sensor nodes working in two modes; the first mode is decision mode; the second mode is transferring mode. We describe the two modes as follows.

(1) Decision Mode. According to "coal mine safety regulation" which is issued by State Administration of Work Safety in China, there are three important values in the monitoring system, which are alarm value, power-off value, and power-recovery value; if the concentration is more than the alarm value, the monitoring system will alert to the workers to prevent the concentration from rising and evacuate from their working place; if the concentration is more than the power-off value, all of the electrical equipment will be powered down to prevent the happening of accident; if the concentration is less than the power-recovery value, the monitoring system will give power to the electrical equipment. So it means some data is not important and some data is important. If the sensor nodes only transfer the important data, it will decrease the power consumption largely. So we propose a threshold value (EV) to help sensor nodes to make a decision; we call this procedure as decision mode. Senor nodes will shut down their communication module, keep collecting environmental parameters, and judge which data should be sent to their neighbor in this mode. When the concentration of CH4 is larger than E, the sensor nodes will be woken up and enter the transferring mode.

(2) Transferring Mode. In this mode, the sensor nodes will send or receive data. Because the structure of monitoring network is shown in Figure 4, the data will only be sent forward and from the sensor node whose ID is larger than another sensor node. Each sensor node can send its data to next sensor node in two hops; to prolong the network life, we propose the cooperation decision mechanism (explained in Section 4.1.2), to help the sensor node decide which node it will send. After that, the sensor node will send the important data to the selected next sensor node until arriving to the sink node.

4. The Data Aggregation Strategy and Prewarning Model

In our study, we propose a data aggregation strategy and a fuzzy comprehensive assessment model based on big data in the coal mine industry.

The data aggregation strategy is used in the process of data transformation from one sensor node to the next sensor node and eventually arrives to the sink node in underground. The fuzzy comprehensive assessment model is used in the prewarning system on the ground to prevent the happening of the accident.

4.1. Data Aggregation Strategy. We propose this strategy that mainly takes into consideration the limited power capacity of the sensor nodes and tends to extend the lifetime of the WSN. First of all, the sensor node collects the data and estimates the importance of the data locally; thus it prohibits communications corresponding to unimportant or

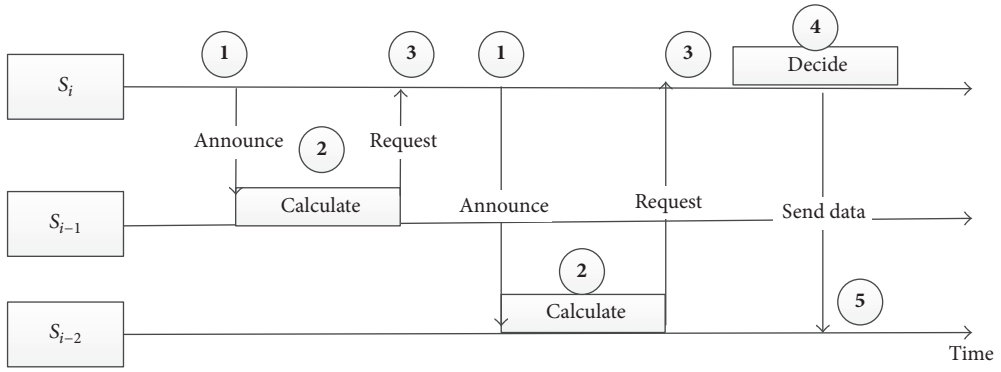


Figure 5: Negotiation process among sensor nodes.

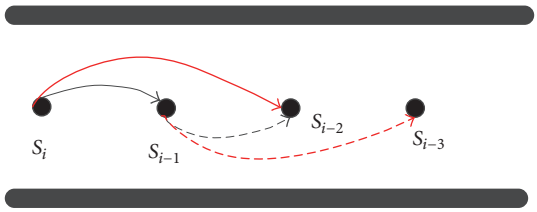


Figure 6: Light load rule.

redundant data. When the sensor node s_i (originator node) detects important data, it will send a message to wake up and ask its closest neighbors in two hops (s_{i-1} and s_{i-2}) to cooperate with it. Neighbors decide to cooperate or not, according to their interests, which are defined by a cooperation decision mechanism (explained in Section 4.1.2). The sensor node s_i (originator node) will choose the best neighbor to cooperate and send data to it. Figure 5 shows the overall negotiation process.

4.1.1. Overall Negotiation Process among Sensor Nodes. As Figure 5 shows, the process consists of five steps. s_i is the first sensor node which detects important data; it will send an announcement message to s_{i-1} and s_{i-2} (step 1); the two neighbors will calculate their cooperate relevance (R) according to cooperation decision mechanism (explained in Section 4.1.2). After their calculation, they will send R to s_i (step 3). s_i receives R and selects the larger one as the next cooperation sensor node (step 4). We assume s_{i-2} has the larger R, so s_i will send data to s_{i-2}.

In some special circumstances, the sensor node should send important data to its neighbor node and receive the announcement message to cooperate with other nodes at the same time. It is shown in Figure 6 that the sensor nodes s_{i-1} and s_i detect important data, so s_{i-1} needs not only to send but also receive the announcement message. In this condition, we will have a rule that the data should always be sent to the light load sensor node. It means s_i will send data to s_{i-2} and s_{i-1} will send data to s_{i-3} (along with the bold line).

4.1.2. Cooperation Decision Mechanism. In this section, we explain the mechanism used in the transferring mode to calculate the cooperate relevance (R) by sensor nodes. When the sensor node wants to send important data to its neighbor nodes, it should choose the better one from them, considered to prolong the lifetime of WSN; we define 4 parameters that may have a large influence on the network. These parameters are as follows: the energy (E), the density (D), the position (P), and the data important degree (I). We use (1) [24] to calculate R. These parameters will be explained in detail later in this section.

$$R = E \times \delta_e + \frac{1}{D} \times \delta_d + P \times \delta_p + I \times \delta_i, \tag{1}$$

where δ_e, δ_d, δ_p, and δ_i are the important factors for the energy, the density, the position, and the data important degree, respectively.

(1) Energy. It is the most important parameter in WSN because the sensor nodes are energy limited. If the sensor node is power exhausted, it will decrease the lifetime of WSN. So we use E to represent the residual energy level. If the sensor node has more residual energy, it is advised to participate in cooperation; another sensor node will save more energy, so it extends the whole lifetime of WSN. We use (2) to calculate E; E_r refers to the residual energy; E_0 refers to the initial energy.

$$E = \frac{E_r}{E_0}. \tag{2}$$

(2) Density. The density is the number of sensor nodes per square meter. We will consider the number of neighbor sensor nodes within its radio range according to [31]. If the sensor node has more neighbors within its radio range, it means the value of D is bigger, the distance between two sensor nodes is shorter, and the energy consumption will be less, so it is advised to participate in cooperation. That is why, in (1), we take the inverse of the density (D) to calculate. We define the density (D) with

$$D = \frac{N_r / (\pi \times r^2)}{N_{\text{ideal}} / (\pi \times r^2)} = \frac{N_r}{N_{\text{ideal}}} \tag{3}$$

where r refers to the radio range of the sensor node, N_r refers to the number of sensor nodes within the radio range, and

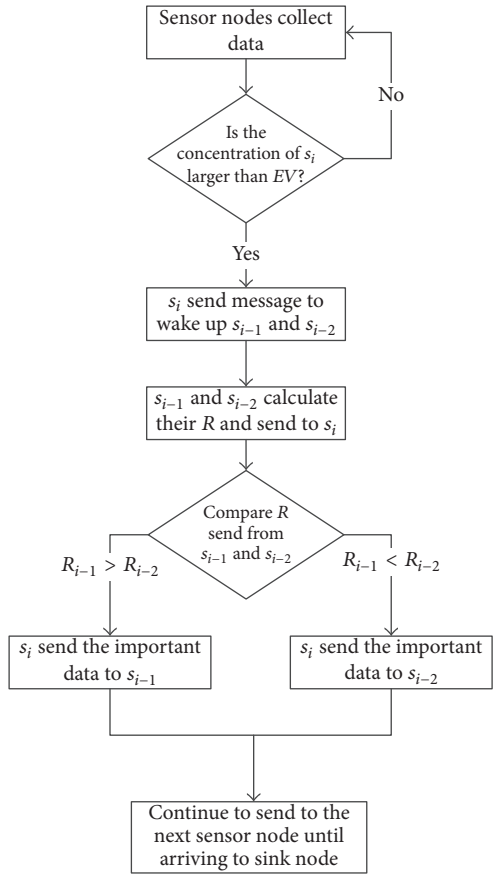


Figure 7: The process of data aggregation strategy.

N_{ideal} refers to theoretical number of sensor nodes and it is given from the ideal distribution of sensor nodes according to [31]. In the ideal case, N_r should be equal to N_{ideal}.

(3) Position. There are two types of positions in WSN. The first position is the normal position, where the sensor node has multiple neighbors. The second position is the edge sensor node, which stays at the edge of the network. In fact, only two sensor nodes are belonging to the second position; they are the sensor nodes which have the largest ID and the smallest ID. We define the position as the distance between the sensor node with sink node in the same coal working face.

(4) Data Important Degree. This parameter depends on the running application and the trend of concentration changing of CH4. It is calculated by local processing, where the sensor node estimates the data important degree according to the rule. For example, if the increment of concentration is growing in 6 hours, it indicates some dangerous thing will happen even the concentration is less than the threshold value (EV). The data will be estimated as important; otherwise, the data is considered unimportant. Figure 7 describes the process of data aggregation strategy.

4.2. Fuzzy Comprehensive Assessment Model. There are many types of sensor nodes in the coal mine. We focus on the concentration of CH4 in the monitoring system, but there are numerous factors that may give influence to the safety of coal mine, because the environment underground is very complicated. If we use single-layer evaluation model in the prewarning system, the incorrect divide will happen because some key parameters may be neglected. It is similar with the mode proposed by [45], so we use it as a reference and propose our fuzzy comprehensive assessment model.

According to "coal mine safety regulation," the different place has different requirements. Coal working face and return airway are the most dangerous places; most explosions and roof accidents have happened in this place. The main haulage roadways are relatively safe because the air is fresh and the geological conditions are better than other places. Reference [36] sums up and analyzes the main reasons of gas, roof, transportation, floods, and fire accident, so we propose a fuzzy comprehensive assessment model in this section.

The monitoring system includes environmental monitoring, equipment operation status monitoring, coal mine transportation monitoring, and so on. We take the environmental monitoring system as an example; it mainly includes the concentration of CH_4, CO, O_2, C_2H_2, and so on.

Assuming the state factor set of environmental monitoring in the coal mine is $U = \{u_1, u_2, \ldots, u_n\}$, u_i $(1 \leqslant i \leqslant n)$ is a certain evaluation factor of the environment. The evaluation set of the safety prewarning model is $X = \{x_1, x_2, \ldots, x_m\}$; x_i $(1 \leqslant j \leqslant m)$ is the alert level of the coal mine. So the evaluation model is the equal of structuring a mapping rule $f : U \to X$, making the only sure comments $X_0 = \{x_1', x_2', \ldots, x_m'\}$ that correspond to the facts $U_0 = \{u_1', u_2', \ldots u_n'\}$. So, the safety rewarning evaluation model considers various factors and gets the alert level x_k $(1 \leqslant k \leqslant m)$.

By using the multilayer comprehensive evaluation model, the state factor set U is divided into h subsets; we call them $U = \{u_1, u_2, \ldots, u_h\}$ and assume the corresponding evaluation weight matrix is $W = \{w_1, w_2, \ldots, w_h\}$, in which W_i $(i = 1, 2, \ldots, h)$ is the weight set of each factor subset. Each element is weight set and satisfies the normalization condition $\sum_{i=1}^{h} W_i = 1$. R $(j = 1, 2, \ldots, h)$ express the fuzzy constraint relationship between each factor subset and evaluation set.

At the beginning, from the first layer, the level 1 assessment $G_i = W_i \circ R_i$ $(i = 1, 2, \ldots, h)$. Then assemble the G_i as the level 2 fuzzy constraint relationship R between factor set and evaluation set. Then step by step evaluate the level set according to the process until arriving to the highest level. The two-layer fuzzy comprehensive evaluation model is shown in Figure 8. The evaluation flow is shown in Figure 9.

5. Performance Evaluation

5.1. Simulations. In order to confirm the performance of data aggregation strategy and fuzzy comprehensive assessment model, we use MATLAB and GlomoSim [37, 43, 44] to do some simulations.

In Table 1, we choose the simulation parameters as close as possible to the reality. For example, the sensor nodes

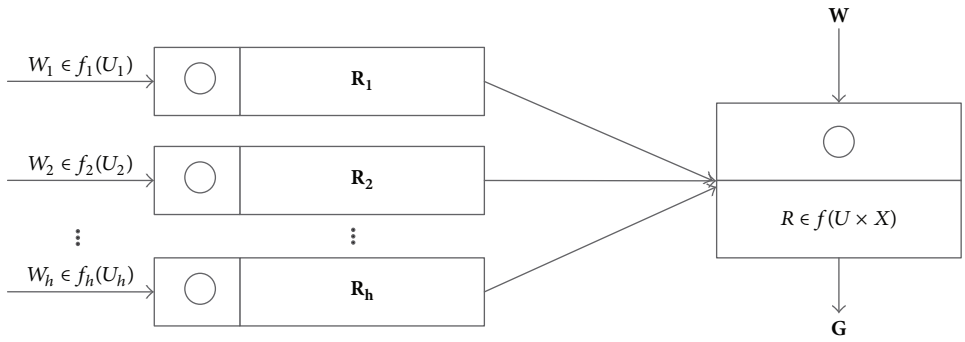


Figure 8: Two-layer fuzzy comprehensive evaluation model.

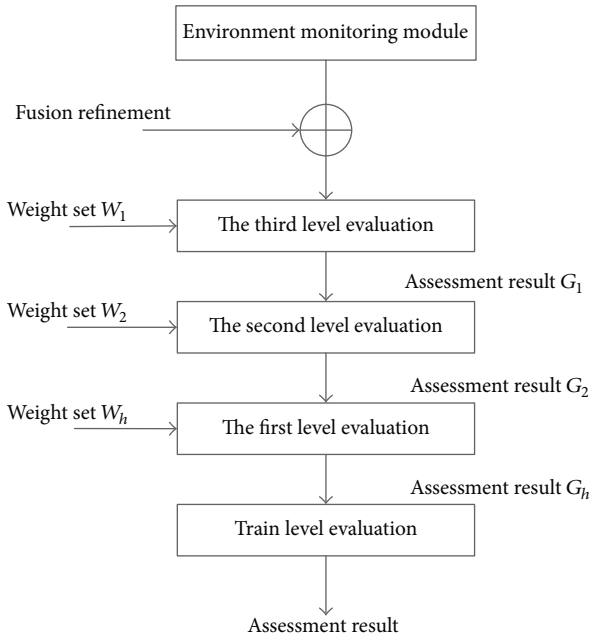


Figure 9: The evaluation flow.

characteristics (transmission, processor, radio range, etc.) are determined according to the specification of [31, 35, 39, 42].

Table 2 resumes the values of importance factor δ_e, δ_p, δ_i, and δ_d. By giving the same value to δ_e, δ_p, and δ_i, it means we give the same importance to the energy, the position, and the data importance degree to calculate the cooperate relevance (R). After we make several simulations, we have found that best values for δ_e and δ_d are 0.35 and δ_d is 0.15. According to the "coal mine safety regulation" and simulations, the threshold value (EV) of CH4 should be between 0.5 and 1.5; for the balance of produce with safety, we choose to fix a threshold of EV to 0.7.

5.2. Performance Analysis. In this section, we will compare and analyze the energy consumption, the delay, the cooperation cost, and the prewarning time. The goal of data aggregation strategy is to reduce the amount of communicated data, increase the valid data in big data, and hence reduce the energy consumption and prolong the lifetime of the network.

Table 1: Simulation parameters.

Simulation parameters	Values
Network coverage	(0, 0)–(1000, 20) m
Number of sink nodes	1–6
Number of sensor nodes	0–600
Initial energy of sensor node	0.5 J
Radio range	75 m
Throughput	1 Mbps
Simulation time	48 hours

Table 2: Cooperate relevance equation parameters.

Equation parameters	Values
δ_e, δ_p, δ_i	0.35
δ_d	0.15

The aim of fuzzy comprehensive assessment model is to decrease the time of prewarn.

In the simulation, we compare the data aggregation strategy with PCEB-MS protocol which is proposed in our previous work [35] and give a simulation to prove the fuzzy comprehensive assessment model can reduce the prewarning time compared with the other methods [33].

(1) Energy Consumption. We define the energy consumption as the average value of the energy consumed by each sensor node during their transmit, receive, and process data. We use the classical energy consumption model [39–41] for simulation. Figure 10 shows the comparison between PCEB-MS and our strategy. The result proves that our strategy is significantly better than our previous work. At the beginning, the average consumption of two methods is similar, but with the increase of sensor nodes, our strategy can save more energy; when the number of sensor nodes is 600, the average energy consumption of PCEB-MS is 15×10^4, but our strategy is only about 10×10^4, so the average energy consumption is reduced by about 34%.

(2) Delay. We define the delay as the average latency needed to send a message from a sensor node which detects important data to the sink; it is including the communication and processing time. At this simulation, we placed 6 sink nodes.

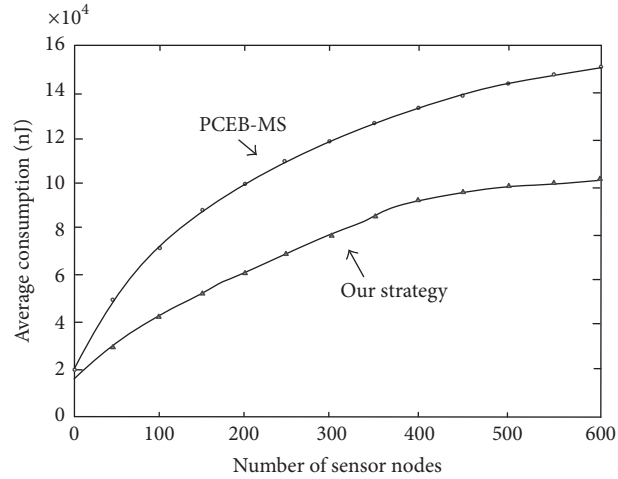


Figure 10: Comparison of average energy consumption.

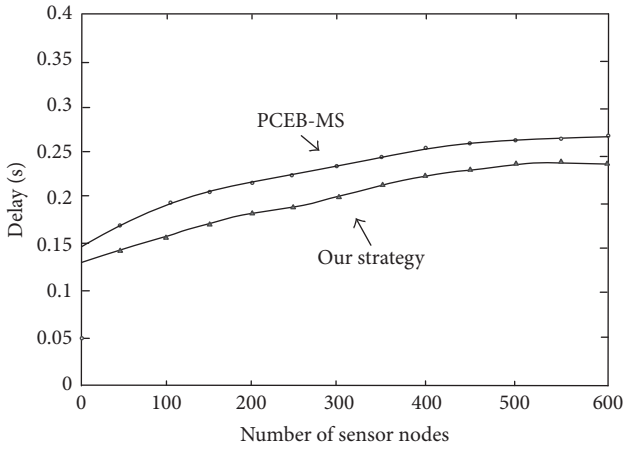


Figure 11: Comparison of delay.

Figure 11 shows our strategy needs less time from each sensor node to sink node because relatively the sensor node does not need process complicated data.

(3) Cooperation Cost. When the sensor node needs to send important data, it will wake its neighbor up in two hops and make a decision to choose a neighbor to send data, so it will cause cooperation cost. Figure 12 shows with the increase of the sensor nodes the cooperation cost will decrease and the consumption is very little. It means the cost is valuable.

(4) Prewarning Time. We use the prewarning time to make sure the fuzzy comprehensive assessment model is reasonable. We define it as how long the time is before the accident is prewarned. There are some studies about the prewarning system in China, such as [33, 46, 47], but only [33] has the similar prewarning system structure with our study, so we choose [33] as a comparison. Reference [33] has proposed a data analysis platform to forecast the accident through the IOT and big data technology, but it has not used any prewarning model.

Table 3: Comparison result.

Concentration accident	Prewarning time	
	Using our model	Not using our model
CH4	3.5 hours	2.2 hours
CO	2.1 hours	1.5 hours
C_2H^2	2.3 hours	1.8 hours

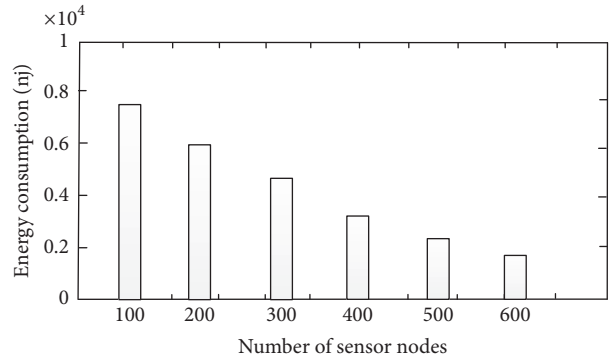


Figure 12: Cooperation cost.

In this simulation, we use the fuzzy comprehensive assessment model on the platform. We get the data from the simulated coal mine, which is a project to train the staff for coal mine industry in our college. Table 3 shows the difference between using this model and not using this model to prewarn of the accident. Obviously, the model can increase about 0.5–1.0 hour of prewarning time.

6. Conclusions

In this paper, we focus on the safety problems of coal mine industry in China. We proposed a monitoring and prewarning system based on big data to prevent happening of the accident. We used the WSN network to replace mine communication cable underground at coal working face and designed the data aggregation strategy and fuzzy comprehensive assessment model to help the system prewarn the accident in advance. At last, we use MATLAB and GlomoSim to do some simulations to make sure our strategy and model are reasonable. Results of simulations show that our method can improve the performance of the monitoring network and prewarning system largely.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61672540, 61379057), 2015 Prevention of Major Accidents Safety Key Technology Projects (Hunan-0012-2015AQ), and 2017 Hunan Natural Science

Foundation of China (2017JJ5004). This job is supported by the Open Program of Xiamen Key Laboratory of Computer Vision and Pattern Recognition, Huaqiao University (no. 600005-Z17X0001) and the Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant no. 20136118120010).

References

[1] W. Haijun and W. Xianli, "Analysis on application of coal mine big data in age of Internet +," *Coal Science & Technology Magazine*, vol. 44, no. 2, pp. 139–143, 2016.

[2] "Big Data and Safety Production Review," http://aqscjdglj.tjftz.gov.cn/system//02/10/010073245.shtml.

[3] X. Xia, Z. Chen, D. Li, and W. Li, "Proposal for efficient routing protocol for wireless sensor network in coal mine goaf," *Wireless Personal Communications*, vol. 77, no. 3, pp. 1699–1711, 2014.

[4] D. Zhang, Z. Chen, M. K. Awad, N. Zhang, H. Zhou, and X. S. Shen, "Utility-optimal resource management and allocation algorithm for energy harvesting cognitive radio sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3552–3565, 2016.

[5] D. Zhang, Z. Chen, H. Zhou, L. Chen, and X. Shen, "Energy-balanced cooperative transmission based on relay selection and power control in energy harvesting wireless sensor network," *Computer Networks*, vol. 104, pp. 189–197, 2016.

[6] E. Sun, X. Zhang, and Z. Li, "The internet of things (IOT) and cloud computing (CC) based tailings dam monitoring and pre-alarm system in mines," *Safety Science*, vol. 50, no. 4, pp. 811–815, 2012.

[7] C. M. Chao and T. Y. Hsiao, "Design of structure-free and energy-balanced data aggregation in wireless sensor networks," *Journal of Network & Computer Applications*, vol. 37, no. 1, pp. 229–239, 2014.

[8] W. Zhao, P. M. Melliar-Smith, and L. E. Moser, "Fault tolerance middleware for cloud computing," in *Proceedings of the 3rd IEEE International Conference on Cloud Computing (CLOUD '10)*, pp. 67–74, Miami, FL, USA, July 2010.

[9] R. Lun and W. Zhao, "A survey of applications and human motion recognition with microsoft kinect," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 29, no. 5, Article ID 1555008, 2015.

[10] "The industry leader in emerging technology research," https://gigaom.com/archives/energy-environment/.

[11] K. Taylor-Sakyi, "Big Data: Understanding Big Data," 2016.

[12] E. Kinoshita and T. Mizuno, *What Is Big Data Big Data Management*, Springer International Publishing, 2017.

[13] C. Bizer, P. Boncz, M. L. Brodie, and O. Erling, "The meaningful use of big data: four perspectives—four challenges," *ACM SIGMOD Record*, vol. 40, no. 4, pp. 56–60, 2011.

[14] D. L. Jones, K. Wagstaff, D. R. Thompson et al., "Big data challenges for large radio arrays," in *Proceedings of the 2012 IEEE Aerospace Conference*, pp. 1–6, Big Sky, MT, USA, March 2012.

[15] F. Yuan, Y. Zhan, and Y. Wang, "Data density correlation degree clustering method for data aggregation in WSN," *IEEE Sensors Journal*, vol. 14, no. 4, pp. 1089–1098, 2014.

[16] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: a tiny aggregation service for ad-hoc sensor networks," *ACM SIGOPS Operating Systems Review*, vol. 36, no. 1, pp. 131–146, 2002.

[17] J. Zheng, P. Wang, and C. Li, "Distributed data aggregation using slepianwolf coding in cluster-based wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 5, pp. 2564–2574, 2010.

[18] M. C. Vuran, Ö. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: theory and applications for wireless sensor networks," *Computer Networks*, vol. 45, no. 3, pp. 245–259, 2004.

[19] J. Yuan and H. Chen, "The optimized clustering technique based on spatial-correlation in wireless sensor networks," in *Proceedings of the 2009 IEEE Youth Conference on Information, Computing and Telecommunication (YC-ICT '09)*, pp. 411–414, Beijing, China, September 2009.

[20] A. Rajeswari and P. T. Kalaivaani, "Energy efficient routing protocol for wireless sensor networks using spatial correlation based medium access control protocol compared with IEEE 802.11," in *Proceedings of the 2011 International Conference on Process Automation, Control and Computing (PACC '11)*, pp. 1–6, Coimbatore, India, July 2011.

[21] J. N. Al-Karaki, R. Ul-Mustafa, and A. E. Kamal, "Data aggregation and routing in wireless sensor networks: optimal and heuristic algorithms," *Computer Networks*, vol. 53, no. 7, pp. 945–960, 2009.

[22] C. Hua and T.-S. P. Yum, "Optimal routing and data aggregation for maximizing lifetime of wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 4, pp. 892–903, 2008.

[23] Z. Sun, H. Song, H. Wang, and X. Fan, "Energy balance-based steerable arguments coverage method in WSNs," *IEEE Access*, p. 99, 2017.

[24] A. Sardouk, M. Mansouri, L. Merghem-Boulahia, D. Gaïti, and R. Rahim-Amoud, "Multi-agent system based wireless sensor network for crisis management," in *Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM '10)*, pp. 1–6, University of Kansas, January 2011.

[25] M. Zhang and M. Shen, "Research of WSN-based data fusion in water quality monitoring," *Computer Engineering & Applications*, vol. 50, no. 23, pp. 234–238, 2014.

[26] W. Wei, Q. Xu, L. Wang et al., "GI/Geom/1 queue based on communication model for mesh networks," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 3013–3029, 2014.

[27] F. Wang, X. Zhang, M. Wang, and G. Chen, "Energy-efficient routing algorithm for WSNs in underground mining," *Journal of Networks*, vol. 7, no. 11, pp. 1824–1829, 2012.

[28] W. Chen, X. Jiang, X. Li, J. Gao, X. Xu, and S. Ding, "Wireless Sensor Network nodes correlation method in coal mine tunnel based on Bayesian decision," *Measurement*, vol. 46, no. 8, pp. 2335–2340, 2013.

[29] G. Zhou, L. Huang, and Z. Zhu, "A Zoning Strategy for Uniform Deployed Chain-Type Wireless Sensor Network in Underground Coal Mine Tunnel," in *Proceedings of the 2013 IEEE 10th International Conference on High Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC)*, pp. 1135–1138, Zhangjiajie, China, November 2013.

[30] W. Wei, X.-L. Yang, P.-Y. Shen, and B. Zhou, "Holes detection in anisotropic sensornets: topological methods," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 135054, 9 pages, 2012.

[31] W. Farjow, K. Raahemifar, and X. Fernando, "Novel wireless channels characterization model for underground mines,"

Applied Mathematical Modelling, vol. 39, no. 19, pp. 5997–6007, 2015.

[32] S. Latif and X. Fernando, "A greener MAC layer protocol for smart home wireless sensor networks," in *Proceedings of the 2013 IEEE Online Conference on Green Communications (OnlineGreenComm '13)*, pp. 169–174, Piscataway, NJ, USA, October 2013.

[33] C.-M. Li, R. Nie, and X.-Y. Qian, "Forecast and prewarning of coal mining safety risks based on the internet of things technology and the big data technology," *Electronic Journal of Geotechnical Engineering*, vol. 20, no. 20, pp. 11579–11586, 2015.

[34] Y. Zhang, W. Yang, D. Han, and Y.-I. Kim, "An integrated environment monitoring system for underground coal mines-Wireless Sensor Network subsystem with multi-parameter monitoring," *Sensors*, vol. 14, no. 7, pp. 13149–13170, 2014.

[35] X. Xia, Z. Chen, H. Liu, H. Wang, and F. Zeng, "A routing protocol for multisink wireless sensor networks in underground coalmine tunnels," *Sensors*, vol. 16, no. 12, pp. 2032–2054, 2016.

[36] S. Jiping, "Accident analysis and big data and Internet of Things in coal mine," *Industry and Mine Automation*, vol. 41, no. 3, pp. 1–5, 2015.

[37] V. Mishra and S. Jangale, "Analysis and comparison of different network simulators," *International Journal of Application or Innovation in Engineering & Management*, 2014.

[38] Y. Liu, A. Liu, S. Guo, Z. Li, Y. Choi, and H. Sekiya, "Context-aware collect data with energy efficient in Cyber–physical cloud systems," *Future Generation Computer Systems*, 2017.

[39] X. Liu, G. Li, S. Zhang, and A. Liu, "Big program code dissemination scheme for emergency software-define wireless sensor networks," *Peer-to-Peer Networking and Applications*, pp. 1–22, 2017.

[40] X. Chen, M. Ming, and L. Anfeng, "Dynamic Power Management and Adaptive Packet Size Selection for IoT in e-Healthcare," *Computers & Electrical Engineering*, 2017.

[41] X. Fan, H. Song, and X. Fan, "Imperfect information dynamic stackelberg game based resource allocation using hidden markov for cloud computing," *IEEE Transactions on Services Computing*, no. 99, p. 1, 2016.

[42] N. Zhang, H. Chen, X. Chen, and J. Chen, "ELM meets urban computing: ensemble urban data for smart city application," in *Proceedings of the ELM-2015*, vol. 1, pp. 51–63, 2016.

[43] W. Zhao, "A Byzantine Fault Tolerant Distributed Commit Protocol," in *Proceedings of the Third IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC '07)*, pp. 37–46, Columbia, MD, USA, September 2007.

[44] H. Song, W. Li, and P. Shen, "Gradient-driven parking navigation using a continuous information potential field based on wireless sensor network," *Information Sciences*, vol. 408, pp. 100–114, 2017.

[45] S. Hou, F. Dou, Y. Li, and Z. Long, "Assessment model of the maglev train braking system safety pre-warning and the optimization of Parameters," in *Proceedings of the 28th Chinese Control and Decision Conference (CCDC '16)*, pp. 4915–4920, May 2016.

[46] L. I. Hao-Min, L. U. Jian-Jun, and C. Wei, "Research of coal mine safety monitoring and early warning system based on cloud computing," *Industry and Mine Automation*, vol. 39, no. 3, pp. 46–50, 2013.

[47] C. Qinggui, Z. Jing, S. Qihua, and Y. Kai, "Design and Application of Hidden Danger Management and Early—warning System for Coal Mine Accidents," *Mining Safety & Environmental Protection*, vol. 43, no. 3, pp. 107–114, 2016.

Hindawi
Scientific Programming
Volume 2018, Article ID 2943290, 7 pages
https://doi.org/10.1155/2018/2943290

Research Article

Developing a Novel Hybrid Biogeography-Based Optimization Algorithm for Multilayer Perceptron Training under Big Data Challenge

Xun Pu,[1] ShanXiong Chen,[1] XianPing Yu,[1] and Le Zhang [1,2]

[1]*College of Computer & Information Science, Southwest University, Chongqing, China*
[2]*College of Computer Science, Sichuan University, Chengdu, China*

Correspondence should be addressed to Le Zhang; zhangle06@scu.edu.cn

Received 24 August 2017; Revised 8 December 2017; Accepted 18 January 2018; Published 1 March 2018

Academic Editor: Anfeng Liu

A Multilayer Perceptron (MLP) is a feedforward neural network model consisting of one or more hidden layers between the input and output layers. MLPs have been successfully applied to solve a wide range of problems in the fields of neuroscience, computational linguistics, and parallel distributed processing. While MLPs are highly successful in solving problems which are not linearly separable, two of the biggest challenges in their development and application are the local-minima problem and the problem of slow convergence under big data challenge. In order to tackle these problems, this study proposes a Hybrid Chaotic Biogeography-Based Optimization (HCBBO) algorithm for training MLPs for big data analysis and processing. Four benchmark datasets are employed to investigate the effectiveness of HCBBO in training MLPs. The accuracy of the results and the convergence of HCBBO are compared to three well-known heuristic algorithms: (a) Biogeography-Based Optimization (BBO), (b) Particle Swarm Optimization (PSO), and (c) Genetic Algorithms (GA). The experimental results show that training MLPs by using HCBBO is better than the other three heuristic learning approaches for big data processing.

1. Introduction

The term big data [1–3] had been developed to describe the phenomenon of the increasing size of massive datasets in scientific experiments, financial trading, and networks. Since big data is always of big volume and has multiple varied types and fast update velocity [4], it is urgent for us to develop such a tool that can extract the meaningful information from big data. Neural networks (NNs) [5, 6] are one of popular machine learning computational approaches, which are composed of several simple and interconnected processing elements and good at loosely modeling the neuronal structures of the human brain. A neural network can be represented as a highly complex nonlinear dynamic system [5], which has some unique characteristics: (a) high dimensionality, (b) extensive interconnectivity, (c) adaptability, and (d) ability to self-organize.

In the last decade, feedforward neural networks (FNNs) [6] have gained popularity in various areas of machine learning [7] and big data mining [1] to solve classification and regression problems. While the two-layered FNN is the most popular neural network used in practical applications, it is not suitable for solving nonlinear problems [7, 8]. The Multilayer Perceptron (MLP) [9, 10], a feedforward neural network with one or more hidden layers between the input and the output layers, is more successful in dealing with nonlinear problems such as pattern classification, big data prediction, and function approximation. Previous research [11] shows that MLPs with one hidden layer are able to approximate any continuous or discontinuous function. Therefore, the study of MLPs with one hidden layer has gained a lot of attention from the research community.

Theoretically, the goal of the learning process of MLPs is to find the best combination of weights and biases of the connections in order to achieve minimum error for the given train and test data. However, one of the most common problems of training an MLP is that there is a tendency for the algorithm to converge on a local minimum. Since an MLP can consist of multiple local minima, it is easy to be trapped in

one of them rather than converging on the global minimum. This is a common problem in most gradient-based learning approaches such as backpropagation (BP) based NNs [12]. According to Mirjalili's research [13], the initial values of the learning rate and the momentum can also affect the convergence in case of BP based NNs, with unsuitable values for these variables resulting in their divergence. Thus, many studies focus on using novel heuristic optimization methods or evolutionary algorithms to resolve the problems of MLP learning algorithms [14]. Classical applied approaches are Particle Swarm Optimization (PSO) algorithms [15, 16], Ant Colony Optimization (ACO) [17], and Artificial Bee Colony (ABC) [18]. However, the No Free Lunch (NFL) theorem [19, 20] states that no heuristic algorithm is best suited for solving all optimization problems. Most of them have their own side effects and overall there has been no significant improvement [13] using these approaches. For example, Genetic Algorithms (GA) may reduce the probability of getting trapped in a local minimum, but they still suffer from slow convergence rates.

Recently, a novel optimization method called Biogeography-Based Optimization (BBO) [21] has been proposed. It is based on the motivation that geographical distribution of biological organisms can be represented by mathematical equations. It is a distributed paradigm, which seeks to simulate the collective behavior of unsophisticated individuals interacting locally with their environment to efficiently identify optimum solutions in complex search spaces. There are many related works of research [22–25] which show that the BBO algorithm is a type of evolutionary algorithm which can offer a specific evolutionary mechanism for each individual in a population. This mechanism makes the BBO algorithm more successful and robust on nonuniform training procedures than gradient-based algorithms. Moreover, compared with the PSO or ACO, the mutation operator of the BBO algorithm can enhance their exploitation capability. This allows the BBO algorithm to outperform PSOs in training MLPs. This has led to a great interest in applying the efficiency of BBO in training MLPs. In 2010, Ovreiu and Simon [24] trained a neuro-fuzzy network with BBO for classifying P-wave features for the diagnosis of cardiomyopathy. Research [13] used 11 standard datasets to provide a comprehensive test bed for investigating the abilities of the BBO algorithm in training MLPs. In this paper, we propose a hybrid BBO with chaotic maps trainer (HCBBO) for MLPs. Our approach employs chaos theory to improve the performance of the BBO with very little computational burden. In our algorithm, the migration and mutation mechanisms are combined to enhance the exploration and exploitation abilities of BBO, and a novel migration operator is proposed to improve BBO's performance in training MLPs.

The rest of this paper is organized as follows. In Section 2, a brief review of the MLP notation and a simple first-order training method are provided. In Sections 3 and 4, the HCBBO framework is introduced and analyzed. In Section 5, the computational results to demonstrate the effectiveness of the proposed improved hybrid algorithm are provided. Finally, Section 6 provides concluding remarks and suggests some directions for future research.

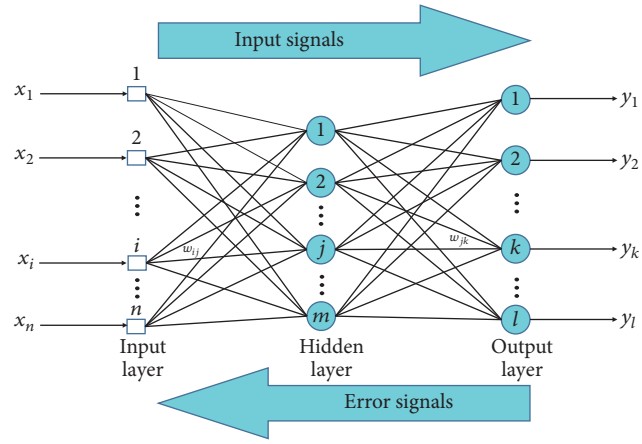


Figure 1: An MLP with one hidden layer.

2. Review of the MLP Notation

The notation used in the rest of the paper represents a fully connected feedforward MLP network with a single hidden layer (as shown in Figure 1). This MLP consists of an input layer, an output layer, and a single hidden layer. The MLP is trained using a backpropagation (BP) learning algorithm. Let n denote the number of input nodes, m denote the number of hidden nodes, and l denote the number of output nodes. Let the input weights $w_{i,j}$ connect the ith input to the jth hidden unit and output weights $w_{\text{out}(j,k)}$ connect the jth hidden unit to the kth output. The weighted sums of inputs are first calculated by the following equation:

$$s_j = \sum_{i=1}^{n} \left(w_{ij} x_i\right) - \theta_j, \quad j = 1, 2, \ldots, m, \tag{1}$$

where n is the number of the input nodes, w_{ij} is the connection weight from the ith node in the input layer to the jth node in the hidden layer, x_i indicates the ith input, and θ_j means the threshold of the jth hidden node.

The output of each hidden node is calculated as follows:

$$f(j) = \frac{1}{\left(1 + \exp\left(-s_j\right)\right)} \quad j = 1, 2, \ldots, m. \tag{2}$$

After calculating outputs of the hidden nodes, the final output can be defined as follows:

$$o_k = \sum_{j=1}^{m} W_{jk} \cdot f(j) - \theta'_k \quad k = 1, 2, \ldots, l, \tag{3}$$

where W_{jk} is the connection weight from the jth hidden node to the kth output node and θ'_k is the bias of the kth output node.

The learning error E (fitness function) is calculated as follows:

$$E_k = \sum_{i=1}^{l} \left(o_i^k - d_i^k\right)^2,$$
$$E = \sum_{k=1}^{q} \frac{E_k}{q}, \tag{4}$$

where q is the number of training samples, l is the number of outputs, d_i^k is the desired output of the ith input unit when the kth training sample is used, and o_i^k is the actual output of the ith input unit when the kth training sample is used.

From the above equations, it can be observed that the final value of the output in MLPs depends upon the parameters of the connecting weights and biases. Thus, training an MLP can be defined as the process of finding the optimal values of the weights and biases of the connections in order to achieve the desirable outputs from certain given inputs.

3. The Proposed Hybrid BBO for Training an MLP

Biogeography-Based Optimization (BBO) is a population-based optimization algorithm inspired by evolution and the balance of predators and preys in different ecosystems. Experiments show that results obtained using the BBO are at least competitive with other population-based algorithms. It has been shown to outperform some well-known heuristic algorithms such as PSO, GA, and ACO on some real-world problems and benchmark functions [21].

The steps of the BBO algorithm can be described as follows. In the beginning, the BBO generates a random number of search agents named habitats, which are represented as vectors of the variables in the problem (analogous to chromosomes in GA). Next, each agent is assigned emigration, immigration, and mutation rates which simulate the characteristics of different ecosystems. In addition, a variable called HSI (the habitat suitability index) is defined to measure the fitness of each habitat. Here, a higher value of HSI indicates that the habitat is more suitable for the residence of biological species. In other words, a solution of the BBO with a high value of HSI indicates a superior result, while a solution with a low value of HSI indicates an inferior result.

During the course of iterations, a set of solutions is maintained from one iteration to the next, and each habitat sends and receives habitants to and from different habitats based on their immigration and emigration rates which are probabilistically adapted. In each iteration, a random number of habitants are also occasionally mutated. That makes each solution adapt itself by learning from its neighbors as the algorithm progresses. Here, each solution parameter is denoted as a suitability index variable (SIV).

The process of BBO is composed of two phases: migration and mutation. During the migration phase, immigration (λ_k) and emigration (μ_k) rates of each habitat follow the model as depicted in Figure 2. A high number of habitants in a habitat increase the probability of emigration and decrease the probability of immigration. During the mutation phase, the mutation factor in BBO keeps the distribution of habitants in a habitat as diverse as possible. In contrast with the mutation factor in GA, the mutation factor of BBO is not set randomly; it is dependent on the probability of the number of species in each habitat.

The mathematical formula of immigration (λ_k) and emigration (μ_k) can be written as follows:

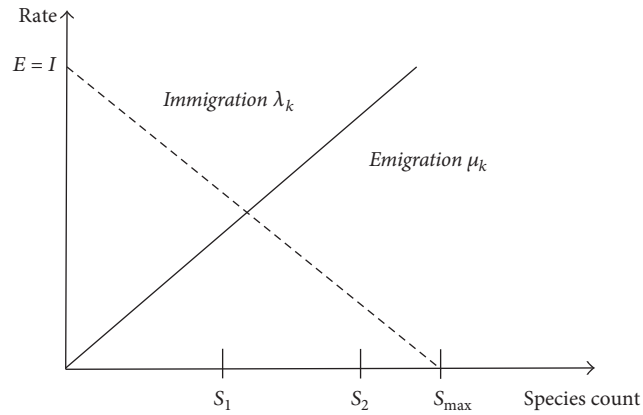


Figure 2: Species model of a habitat.

$$\lambda_k = I\left(1 - \frac{S_k}{S_{\max}}\right),$$
$$\mu_k = E\left(\frac{S_k}{S_{\max}}\right), \tag{5}$$

where I is the maximum immigration rate, E is the maximum emigration rate, $S_{\max}$ is the maximum number of habitants, and S_k is the habitant count of k.

The mutation of each habitat, which improves the exploration of BBO, is defined as follows:

$$m(s) = m_{\max} \times \left(1 - \frac{P_n}{P_{\max}}\right). \tag{6}$$

Here $m_{\max}$ is the maximum value of mutation defined by user, $P_{\max}$ is the greatest mutation probability of all the habitats, and P_n is the mutation probability of the nth habitat, which can be obtained as

$$\dot{P}_n = \begin{cases} -(\lambda_n + \mu_n) P_n + \mu_{n+1} P_{n+1}, & n = 0; \\ -(\lambda_n + \mu_n) P_n + \mu_{n+1} P_{n+1} + \lambda_{n-1} P_{n-1}, & 1 \le n \le S_{\max} - 1 = 0; \\ -(\lambda_n + \mu_n) P_n + \lambda_{n-1} P_{n-1}, & n = S_{\max}. \end{cases} \tag{7}$$

The complete process of the BBO algorithm is described in Algorithm 1; here $I : \phi \rightarrow \{H^n, \mathrm{HSI}^n\}$ initializes an ecosystem of habitats and computes each corresponding HSI and $\Gamma = (n, m, \lambda, \tau, \Omega, M)$ is a transition function which modifies the ecosystem from one optimization iteration to the next. The elements of the 6-tuple can be defined as follows: n is the number of habitats; m is the number of SIVs; λ is the immigration rate; τ is the emigration rate; Ω is the migration operator; and M is the mutation operator.

4. The Proposed Hybrid CBBO Algorithm for Training an MLP

There are three different approaches for using heuristic algorithms for training MLPs. In the first approach, heuristic algorithms are employed to find a combination of weights

$I : \phi \rightarrow \{H^n, \mathrm{HSI}^n\}$
While (condition = T)
    $\Gamma = (n, m, \lambda, \tau, \Omega, M)$
end

Algorithm 1: Pseudocode of BBO for optimization problems.

and biases to provide the minimum error for an MLP. In the second approach, heuristic algorithms are utilized to find the proper architecture for an MLP to be applied to a particular problem. In the third approach, heuristic algorithms can be used to tune the parameters of a gradient-based learning algorithm.

Mirjalili et al. [13] employed the basic BBO algorithm to train an MLP using the first approach, and the results demonstrate that BBO is significantly better at avoiding local minima compared to PSO, GA, and ACO algorithms. However, the basic BBO algorithm still has some drawbacks, such as (a) the large number of iterations needed to reach the global optimal solution and (b) the tendency to converge to solutions which may be locally the best. Many methods have been proposed to improve the capabilities for the exploration and exploitation of the BBO algorithm.

4.1. Chaotic Systems. Chaos theory [26] refers to the study of chaotic dynamical systems, which is embodied by the so-called "butterfly effect." As nonlinear dynamical systems, chaotic systems are highly sensitive to their initial conditions, and tiny changes to their initial conditions may result in significant changes in the final outcomes of these systems.

In this paper, chaotic systems are applied to BBOs instead of random values [25–27] for their initialization. This means that chaotic maps substitute the random values to provide chaotic behaviors to heuristic algorithms. During the processing of the BBO algorithm, the most important random values are calculated to choose a habitat for emigrating the new habitants during the migration phase. We utilize chaotic maps, which use the logistic model in (8), and choose a value from the interval of $[0, 1]$, whenever there is a need for a random value.

$$x_{n+1} = f(x_n) = \frac{\mu}{4} \sin(\pi x_n); \tag{8}$$

here $x_{n+1} \in [0, 1]$ and μ are named logistic parameters. When μ equals 4, the iterations produce values which follow a pseudorandom distribution. This means that a tiny difference in the initial value of x_1 will give rise to a large difference in its long-time behavior. We employ this feature to avoid a local convergence of the BBO algorithm.

4.2. Habitat Suitability Index (Fitness Function). During the training phase of an MLP, each training data sample should be involved in calculating the HIS of each candidate solution. In this work, the Mean Square Error (MSE) is utilized for evaluating all training samples. The MSE is defined as follows:

$$E = \sum_{k=1}^{q} \frac{\sum_{i=1}^{l} \left(o_i^k - d_i^k\right)^2}{q}; \tag{9}$$

here q is the number of training samples, l is the number of outputs, d_i^k is the desired output of the ith input unit when the kth training sample is used, and o_i^k is the actual output of the ith input unit when the kth training sample is used. Thus, the HSI value for the ith candidate is given by $\mathrm{HSI}(c_i) = E(c_i)$.

4.3. Opposition-Based Learning. To improve the convergence of BBO algorithm during the mutation phase, a method named opposition-based learning (OBL) has been used in [22]. The main idea of opposition-based learning is to consider an estimate and its opposite at the same time to achieve a better approximation of the current candidate solution.

Assuming that $X = (x_1, x_2, \ldots, x_n)$ represents a vector of the weights and biases in the MLP, with $x_i \in R$ and $x_i \in [\min_i, \max_i]$ $\forall i \in \{1, 2, \ldots, n\}$, then the definition of the opposite vector is $X' = (x_1', x_2', \ldots, x_n')$ with its elements as $x_i' = \min_i + \max_i - x_i$. The algorithm for the OBL method can be described as follows:

(1) Generate a vector $X = (x_1, x_2, \ldots, x_n)$ and its opposite $X' = (x_1', x_2', \ldots, x_n')$, in an n-dimensional search space.

(2) Evaluate the fitness of both points, $\mathrm{HSI}(X)$ and $\mathrm{HSI}(X')$.

(3) If $\mathrm{HSI}(X) \leq \mathrm{HSI}(X')$, then replace X with X'; otherwise, continue with X.

Thus, the vector and its opposite vector are evaluated simultaneously to obtain the fitter one.

4.4. Outline of HCBBO for MLP. In this section, the main procedure of HCBBO is described. To guarantee an initial population with a certain quality and diversity, the initial population is generated using a combination of the chaotic system and the OBL approach. By fusing the local search strategies with the migration and mutation phases of the BBO algorithm, the exploration and exploitation capabilities of the HCBBO can be well balanced. The main procedure of our proposed HCBBO to train an MLP can be described as Algorithm 2.

5. Experimental Analysis

This study focuses on finding an efficient training method for MLPs. To evaluate the performance of the proposed HCBBO algorithm in this paper, a series of experiments were developed using the Matlab software environment (V2009). The system configuration is as follows: (a) CPU: Intel i7; (b) RAM: 4 GB; (c) operating system: Windows 8. Based on the works described in [13, 28, 29], we choose four publicly available classification big datasets to benchmark our system: (1) balloon, (2) iris, (3) heart, and (4) vehicle. All these datasets are freely available from the University of California at Irvine (UCI) Machine Learning Repository [30], thus ensuring replicability. And the characteristics of these datasets are listed in Table 1.

(1) **input**: habitat size n, maximum migration rate E and I (emigration and immigration rate), the maximum mutation rate M_{max};
(2) Initialize set of MLPs (habitats) by chaos maps on formula Eq. (8);
(3) For each habitat, calculate its mean square error by relative parameters based on formulas (9). And the basic rule of fitness function is the better performance maintains the smaller value of MSE. Then elite habitats are identified by the values of HSI.
(4) Combing MLPs according to immigration and emigration rates based on Eq. (6)
Probabilistically use immigration and emigration to modify each non-elite habitat based on Eq. (7).
(5) Select number of MLPs and recomputed (mutate) some of their weights or biases by chaos maps.
(6) Save some of the MLPs with low MSE;
(7) This loop will be terminated if a predefined number of generations are reached or an acceptable problem solution has been found, otherwise go to step (3) for the next iteration.
(8) **output**: the MLP with minimum MSE (HSI).

Algorithm 2: The framework of HCBBO algorithm.

Table 1: Classification datasets.

Classification datasets	Number of attributes	Number of training samples	Number of test samples	Number of classes
Balloon	4	16	16 as training samples	2
Iris	4	150	150 as training samples	3
SPECT Heart	22	80	187	2
Vehicle	18	400	446	4

Table 2: The main parameters of BBO and HCBBO.

Maximum number of generations: $T = 300$	Maximum mutation rate: $M_{max} = 0.005$
Elitism parameter: $e = 5$	Maximum possible emigration rate: $E = 1$
Population size: $P_{max} = 200$	Maximum possible immigration rate: $I = 1$

Table 3: MLP structure parameters.

	Balloon	Iris	SPECT Heart	Vehicle
Input	4	4	22	18
Hidden	9	9	45	52
Output	1	3	1	4

In this paper, we compare the performances of 4 algorithms, BBO, PSO, GA, and HCBBO, over the benchmark datasets described in Table 1. Since manually choosing appropriate parameters for each of these algorithms is time-consuming, the initial parameters and property structures for both the classical BBO algorithm and HCBBO algorithms (which were adjusted as Table 2) were chosen as in paper [13].

In order to increase the accuracy of the experiment, each algorithm was run 20 times, and different MLP structures will be used to deal with different datasets, which were listed in Table 3.

The running time (RT) and convergence curves of each algorithm are shown in Figures 3–7. From Figure 3, it can be observed that the average computational time of HCBBO is 8 to 13% lower than the best time obtained for the BBO. It is also lower than the computational time of all the other algorithms compared in this experiment. This decrease in the running time can be attributed to the fact that the HCBBO's search ability was enhanced by OBL.

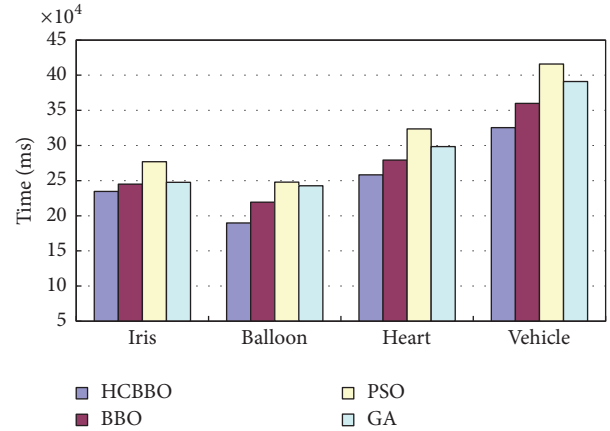


Figure 3: Total running time of each algorithm.

The convergence curves in Figures 4–7 show that, among all the algorithms, HCBBO has the fastest convergence behavior on all the datasets. In Figure 4, under the same experimental conditions, HCBBO achieved the optimal values for its parameters after 150 generations while BBO could not converge to an optimal value even after 200 generations. The same pattern in faster convergence for the HCBBO was observed for the other classical problems (Figures 5–7). Statistically speaking, HCBBO performs the best on all the classification datasets, since it is able to avoid local minima better than any

Table 4: Experimental results for classification rate.

Algorithm	Iris Classification rate	Heart Classification rate	Balloon Classification rate	Vehicle Classification rate
HCBBO	93%	81.2%	100%	76.2%
BBO	90%	75.4%	100%	71.7%
PSO	38%	66.5%	100%	56.8%
GA	88.2%	56.9%	100%	59.9%

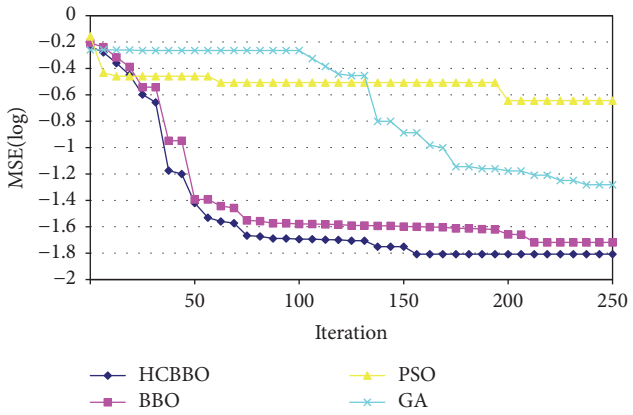


Figure 4: Convergence curves of algorithms for iris dataset.

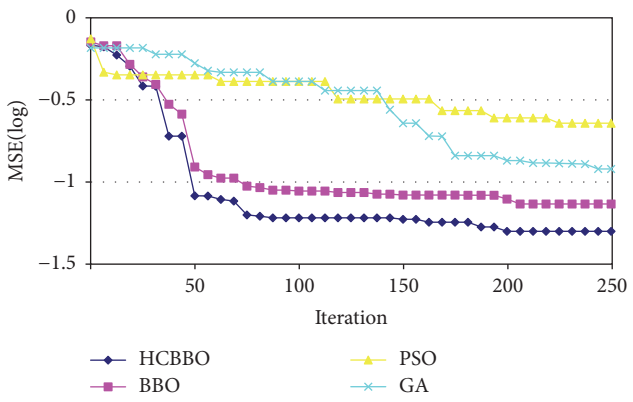


Figure 5: Convergence curves of algorithms for heart dataset.

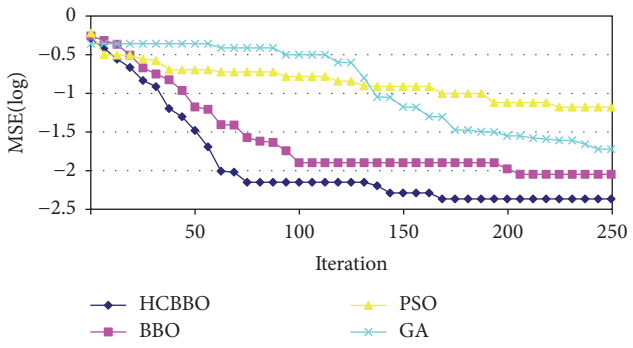


Figure 6: Convergence curves of algorithms for balloon dataset.

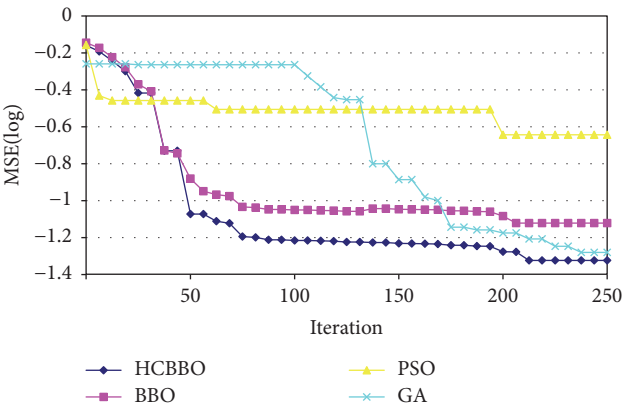


Figure 7: Convergence curves of algorithms for vehicle dataset.

other algorithm. And the classification results obtained by HCBBO are better than all other algorithms for the chosen datasets.

The experimental results of mean classification rate are provided in Table 4. Statistically speaking, HCBBO has the best results in all of the classification datasets because it avoids local minima better.

6. Discussion and Conclusions

In this paper, a HCBBO algorithm was presented for training an MLP. Four benchmark big datasets (balloon, iris, heart, and vehicle) were employed to investigate the effectiveness of HCBBO in training MLPs. The performance results were statistically compared with three state-of-the-art algorithms: BBO, PSO, and GA. The main contributions and innovations of this work are summarized as follows: (a) this is the first research work combining a hybrid chaos system with the BBO algorithm to train MLPs; (b) the method named OBL was used in the mutation operator of HCBBO to improve the convergence of the algorithm; and (c) the results demonstrate that HCBBO has better convergence capabilities than BBO, PSO, and GA. In the future, we will apply the trained neural networks to analyze the big medical data and integrate more novel data mining algorithms [29, 31–35] into HCBBO.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] C. Kacfah Emani, N. Cullot, and C. Nicolle, "Understandable big data: a survey," *Computer Science Review*, vol. 17, pp. 70–81, 2015.

[2] F. J. Alexander, A. Hoisie, and A. Szalay, "Big data [Guest editorial]," *Computing in Science & Engineering*, vol. 13, no. 6, Article ID 6077842, pp. 10–12, 2011.

[3] D. Boyd and K. Crawford, "Critical questions for big data: provocations for a cultural, technological, and scholarly phenomenon," *Information Communication and Society*, vol. 15, no. 5, pp. 662–679, 2012.

[4] P. Hitzler and K. Janowicz, *Linked Data, Big Data, and the 4th Paradigm*, IOS Press, 2013.

[5] B. Irie and S. Miyake, "Capabilities of three-layered perceptrons," in *Proceedings of 1993 IEEE International Conference on Neural Networks (ICNN '93)*, pp. 641–648, San Diego, CA, USA, 1988.

[6] T. L. Fine, "Feedforward neural network methodology," *Information Science & Statistics*, vol. 12, no. 4, pp. 432-433, 1999.

[7] C. W. Deng, G. B. Huang, J. Xu, and J. X. Tang, "Extreme learning machines: new trends and applications," *Science China Information Sciences*, vol. 58, no. 2, pp. 1–16, 2015.

[8] M. H. Hassoun, *Fundamentals of Artificial Neural Networks*, MIT Press, 1995.

[9] R. C. Odom, P. Paul, S. S. Diocee, S. M. Bailey, D. M. Zander, and J. J. Gillespie, "Shaly sand analysis using density-neutron porosities from a cased-hole pulsed neutron system," in *Proceedings of the SPE Rocky Mountain Regional Meeting*, Gillette, Wyoming, 1999.

[10] N. A. Mat Isa and W. M. F. W. Mamat, "Clustered-hybrid multilayer perceptron network for pattern recognition application," *Applied Soft Computing*, vol. 11, no. 1, pp. 1457–1466, 2011.

[11] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[12] D. Rumelhart and J. Mcclelland, *Learning Internal Representations by Error Propagation*, MIT Press, 1988.

[13] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Let a biogeography-based optimizer train your multi-layer perceptron," *Information Sciences*, vol. 269, pp. 188–209, 2014.

[14] A. Van Ooyen and B. Nienhuis, "Improving the convergence of the back-propagation algorithm," *Neural Networks*, vol. 5, no. 3, pp. 465–471, 1992.

[15] I. A. A. Al-Hadi, S. Z. M. Hashim, and S. M. H. Shamsuddin, "Bacterial foraging optimization algorithm for neural network learning enhancement," in *Proceedings of the 2011 11th International Conference on Hybrid Intelligent Systems, HIS 2011*, pp. 200–205, Malaysia, 2011.

[16] V. G. Gudise and G. K. Venayagamoorthy, "Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '03)*, pp. 110–117, Indianapolis, Ind, USA, 2003.

[17] C. Blum and K. Socha, "Training feed-forward neural networks with ant colony optimization: an application to pattern classification," in *Proceedings of the 5th International Conference on Hybrid Intelligent Systems (HIS '05)*, pp. 233–238, 2005.

[18] M. Karacor, K. Yilmaz, and F. Erfan Kuyumcu, "Modeling MCSRM with artificial neural network," in *Proceedings of the 2007 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) and Electromotion '07*, pp. 849–852, Bodrum, Turkey, 2007.

[19] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, no. 237, pp. 82–117, 2013.

[20] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[21] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[22] M. Ergezer and D. Simon, "Oppositional biogeography-based optimization for combinatorial problems," in *Proceedings of the 2011 IEEE Congress of Evolutionary Computation, CEC 2011*, pp. 1496–1503, New Orleans, LA, USA, 2011.

[23] S. S. Malalur, M. T. Manry, and P. Jesudhas, "Multiple optimal learning factors for the multi-layer perceptron," *Neurocomputing*, vol. 149, pp. 1490–1501, 2015.

[24] M. Ovreiu and D. Simon, "Biogeography-based optimization of neuro-fuzzy system parameters for diagnosis of cardiac disease," in *Proceedings of the 12th Annual Genetic and Evolutionary Computation Conference, GECCO-2010*, pp. 1235–1242, New York, NY, USA, 2010.

[25] W. Zhu and H. Duan, "Chaotic predator-prey biogeography-based optimization approach for UCAV path planning," *Aerospace Science and Technology*, vol. 32, no. 1, pp. 153–161, 2014.

[26] S. H. Kellert, "Books-received - in the wake of chaos - unpredictable order in dynamical systems," vol. 267, Science, 95 edition, 1995.

[27] L. Zhang, Y. Xue, B. Jiang et al., "Multiscale agent-based modelling of ovarian cancer progression under the stimulation of the STAT 3 pathway," *International Journal of Data Mining and Bioinformatics*, vol. 9, no. 3, pp. 235–253, 2014.

[28] S. Mirjalili, S. Z. Mohd Hashim, and H. Moradian Sardroudi, "Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm," *Applied Mathematics and Computation*, vol. 218, no. 22, pp. 11125–11137, 2012.

[29] L. Zhang and S. Zhang, "Using game theory to investigate the epigenetic control mechanisms of embryo development: comment on: epigenetic game theory: how to compute the epigenetic control of maternal-to-zygotic transition "by Qian Wang et al"," *Physics of Life Reviews*, vol. 20, pp. 140–142, 2017.

[30] C. J. M. C. Blake, *Repository of Machine Learning Databases*, http://archive.ics.uci.edu/ml/datasets.html.

[31] B. Jiang, W. Dai, A. Khaliq, M. Carey, X. Zhou, and L. Zhang, "Novel 3D GPU based numerical parallel diffusion algorithms in cylindrical coordinates for health care simulation," *Mathematics and Computers in Simulation*, vol. 109, pp. 1–19, 2015.

[32] H. Peng, T. Peng, J. Wen et al., "Characterization of p38 MAPK isoforms for drug resistance study using systems biology approach," *Bioinformatics*, vol. 30, no. 13, pp. 1899–1907, 2014.

[33] Y. Xia, C. Yang, N. Hu et al., "Exploring the key genes and signaling transduction pathways related to the survival time of glioblastoma multiforme patients by a novel survival analysis model," *BMC Genomics*, vol. 18, no. Suppl 1, 2017.

[34] L. Zhang, M. Qiao, H. Gao et al., "Investigation of mechanism of bone regeneration in a porous biodegradable calcium phosphate (CaP) scaffold by a combination of a multi-scale agent-based model and experimental optimization/validation," *Nanoscale*, vol. 8, no. 31, pp. 14877–14887, 2016.

[35] L. Zhang, Y. Liu, M. Wang et al., "EZH2-, CHD4-, and IDH-linked epigenetic perturbation and its association with survival in glioma patients," *Journal of Molecular Cell Biology*, 2017.

Hindawi
Scientific Programming
Volume 2018, Article ID 3732120, 7 pages
https://doi.org/10.1155/2018/3732120

Research Article

An Incremental Optimal Weight Learning Machine of Single-Layer Neural Networks

Hai-Feng Ke,[1] Cheng-Bo Lu,[2] Xiao-Bo Li,[2] Gao-Yan Zhang,[1] Ying Mei,[2] and Xue-Wen Shen[3]

[1]*School of Computer & Computing Science, Zhejiang University City College, Hangzhou 310015, China*
[2]*College of Engineering, Lishui University, Lishui 323000, China*
[3]*School of Electronics and Information, Zhejiang University of Media and Communications, Hangzhou 310015, China*

Correspondence should be addressed to Cheng-Bo Lu; lu.chengbo@aliyun.com

Received 12 October 2017; Revised 1 January 2018; Accepted 11 January 2018; Published 1 March 2018

Academic Editor: Wenbing Zhao

An optimal weight learning machine with growth of hidden nodes and incremental learning (OWLM-GHNIL) is given by adding random hidden nodes to single hidden layer feedforward networks (SLFNs) one by one or group by group. During the growth of the networks, input weights and output weights are updated incrementally, which can implement conventional optimal weight learning machine (OWLM) efficiently. The simulation results and statistical tests also demonstrate that the OWLM-GHNIL has better generalization performance than other incremental type algorithms.

1. Introduction

Feedforward neural networks (FNNs) have been extensively used in classification applications and regressions [1]. As a specific type of FNNs, single hidden layer feedforward networks with additive models can approximate any target continuous function [2]. Owing to excellent learning capabilities and fault tolerant abilities, SLFNs play an important role in practical applications and have been investigated extensively in both theory and application aspects [3–7].

Maybe the most popular training method for SLFNs classifiers in recent years was gradient-based back-propagation (BP) algorithms [8–12]. BP algorithms can be easily recursively implemented in real time; however, the slow convergence is the bottleneck of BP, where the fast training is essential. In [13], a kind of novel learning machine named extreme learning machine (ELM) is proposed for training SLFNs, where the learning parameters of hidden nodes, including input weights and biases, are randomly assigned and need not be tuned, while output weights can be obtained by simple generalized inverse computation. It has been proven that, even without updating the parameters of the hidden layer, SLFNs with randomly generated hidden neurons and tunable output weights maintain their universal approximation and excellent generalization performance [14]. It has been shown that ELM is faster than most state-of-the-art training algorithms for SLFNs and it has been applied widely to many practical cases such as classification, regression, clustering, recognition, and relevance ranking problems [15, 16].

Since input weights and hidden layer biases of SLFNs trained with ELM are randomly assigned, the minor changes of data in input vectors maybe cause large changes of data in hidden layer output matrix of the SLFNs. This in turn will lead to large changes of data in the output weight matrix. According to statistical learning theory [17–21], the large changes of data in the output weight matrix will greatly increase both structural and empirical risks of the SLFNs, which will in turn decrease robustness property of the SLFNs regarding the input disturbances. In fact, it has been noted from simulations that the SLFNs trained with the ELM sometimes perform poor generalization performance and robustness with regard to the input disturbances. In view of this situation, OWLM [22] was proposed; it is seen that both input weights and output weights of the SLFNs are globally optimized with the batch learning type of least squares. All feature vectors of classifier can then be placed at the

prescribed positions in feature space in the sense that the separability of those nonlinearly separable patterns can be maximized, and better generalization performance can be achieved compared with conventional ELM.

However, there is still one major issue existing in OWLM, which is OWLM needing more computational cost than ELM, since the input weights are not randomly selected in SLFNs trained with OWLM. With the advent of the big data age, data sets become larger and more complex [23, 24], which reduces the learning efficiency of OWLM further. For implementing OWLM efficiently, this paper proposed an incremental learning machine referred to as optimal weight learning machine with growth of hidden nodes and incremental learning (OWLM-GHNIL). Whenever new nodes are added, the input weights and output weights could be incrementally updated which can implement the conventional OWLM algorithm efficiently. At the same time, owing to the advantages of OWLM, OWLM-GHNIL has better generalization performance than other incremental algorithms such as EM-ELM [25] (an approach that could automatically determine the number of hidden nodes in generalized single hidden layer feedforward networks) and I-ELM [14], which added random hidden nodes to SLFNs only one hidden node each time.

The rest of this paper is organized as follows: in Section 2, the OWLM is briefly described. In Section 3, we present OWLM-GHNIL in detail and analyze its computational complexity. Simulation results are then presented in Section 4, showing that our proposed approach performs more efficiently and has better generalization performance than some existing methods. In Section 5, we give conclusion.

2. Brief of the Optimal Weight Learning Machine

In this section, we briefly describe the OWLM.

For N given input pattern vectors $x_1(k), x_2(k), \ldots, x_N(k)$, as well as N corresponding desired output data vectors $o_1(k), o_2(k), \ldots, o_N(k)$,respectively, N linear output equations of SLFNs in Figure 1 can be obtained as

$$H\beta = O, \tag{1}$$

where

$$H = \left[x_1(k) \;\; x_2(k) \;\; \cdots \;\; x_N(k)\right]^T W = \begin{bmatrix} x_1^T w_1 & x_1^T w_2 & \cdots & x_1^T w_{\widetilde{N}} \\ x_2^T w_1 & x_2^T w_2 & \cdots & x_2^T w_{\widetilde{N}} \\ \vdots & \vdots & \vdots & \vdots \\ x_N^T w_1 & x_N^T w_2 & \cdots & x_N^T w_{\widetilde{N}} \end{bmatrix} = XW, \tag{2}$$

with

$$X = \left[x_1(k), x_2(k), \ldots, x_N(k)\right]^T, \qquad x_i(k) = \left[x_{i1}(k), x_{i2}(k), \ldots, x_{in}(k)\right] \tag{3}$$

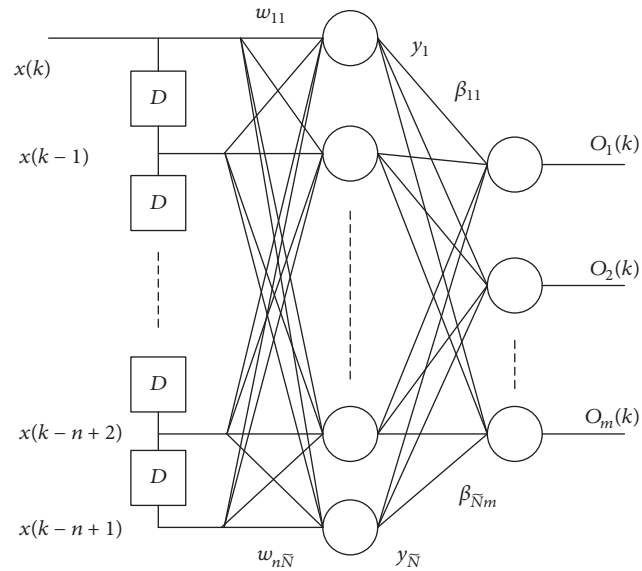


Figure 1: A single hidden layer neural network with linear nodes and an input tapped delay line.

and input weight matrix

$$W = \left[w_1, w_2, \ldots, w_{\widetilde{N}}\right], \qquad w_i = \left[w_{i1}, w_{i2}, \ldots, w_{ni}\right]^T \tag{4}$$

and output weight matrix

$$\beta = \left[\beta_1, \beta_2, \ldots, \beta_m\right], \qquad \beta_i = \left[\beta_{i1}, \beta_{i2}, \ldots, \beta_{i\widetilde{N}}\right]. \tag{5}$$

Let $y_1, y_2, \ldots, y_N$ be N feature vectors, corresponding to the input data vectors $x_1, x_2, \ldots, x_N$. Then we have

$$\left[y_1 \;\; y_2 \;\; y_3 \;\; \cdots \;\; y_N\right] = W^T \left[x_1 \;\; x_2 \;\; x_3 \;\; \cdots \;\; x_N\right], \tag{6}$$

or

$$Y = W^T X. \tag{7}$$

Let the N reference feature vectors be described by

$$Y_d = \left[y_{d1} \;\; y_{d2} \;\; y_{3d} \;\; \cdots \;\; y_{dN}\right]. \tag{8}$$

Generally, as described in [22], the selection of the N desired feature vectors in (8) mainly depends on the characteristics of the input vectors. By optimizing the input weights of the SLFNs in Figure 1, the OWLM can place the feature vectors of the SLFNs at the "desired position" in feature space. The purpose of the assignment is to further maximize the separability of the vectors in the feature space so that the generalization performance and robustness, seen from the output layer of the SLFNs, can be greatly improved, compared with the SLFNs trained with ELM.

The design of the optimal input weight of the SLFNs can be formulated by the following optimization problem:

$$\begin{aligned} &\text{Minimize} \quad \left\{\frac{1}{2}\left\|\varepsilon_f\right\|^2 + \frac{\lambda_1}{2}\|W\|^2\right\} \\ &\text{subject to} \quad \varepsilon_f = Y_d - Y = Y_d - W^T X, \end{aligned} \tag{9}$$

where λ_1 is a positive real regularization parameter.

The optimal input weight matrix was derived as follows:

$$W = \left(\lambda_1 I + XX^T\right)^{-1} XY_d{}^T. \tag{10}$$

Similarly, to minimize the error between the desired output pattern T and the actual output pattern O, the design of the optimal output weight of the SLFNs can be formulated by the following optimization problem:

$$\begin{aligned} &\text{Minimize} \quad \left\{\frac{1}{2}\|\varepsilon\|^2 + \frac{\lambda_2}{2}\left\|\beta\right\|^2\right\} \\ &\text{subject to} \quad \varepsilon = O - T = H\beta - T, \end{aligned} \tag{11}$$

where λ_2 is a positive real regularization parameter.

The optimal output weight matrix was derived as follows:

$$\beta = \left(\lambda_2 I + H^T H\right)^{-1} H^T T. \tag{12}$$

The optimal weight learning machine [22] can be summarized as follows.

Algorithm OWLM. Given a training set $\{(x_i, t_i)\}_{i=1}^N$, as well as hidden node number $\widetilde{N}$, do the following steps.

Step 1. Randomly assign hidden node parameters (w_i, b_i), $i = 1, \ldots, \widetilde{N}$.

Step 2. Calculate the hidden layer output matrix H by (2).

Step 3. Calculate the input weight matrix W_{new} by (10).

Step 4. Recalculate the hidden layer output matrix H_{new} by W_{new}.

Step 5. Calculate the output weight matrix β by (12).

Obviously, the OWLM needs more training time compared with ELM, since it needs additional computational cost for computing the input weight matrix.

3. Growing Hidden Nodes and Incrementally Updating Weights

Given SLFNs with initial hidden nodes m and a training set $\{(x_i, t_i)\}_{i=1}^N$ let N be the number of input patterns, let l be the length of the input patterns, and let h be the length of the output patterns.

We have

$$W_0 = \left(XX^T + \lambda_1 I\right)^{-1} XY_{d0}{}^T, \tag{13}$$

$$H_0 = X^T W_0 \tag{14}$$

$$\beta_0 = \left(H_0{}^T H_0 + \lambda_2 I\right)^{-1} H_0{}^T T, \tag{15}$$

where Y_{d0} is an $m \times N$ matrix consisting of N desired feature vectors.

Let $E(H_i) = \min\|H_i\beta_i - T\|$ be the network output error; if $E(H_0)$ is less than the target error $\varepsilon > 0$, then no new hidden nodes need to be added and the learning procedure completes. Otherwise, we could add n new nodes to the existing SLFNs; then

$$\begin{aligned} W_1 &= \left(XX^T + \lambda_1 I\right)^{-1} X \begin{bmatrix} Y_{d0} \\ Y_{d1} \end{bmatrix}^T \\ &= \left[W_0 \quad \left(XX^T + \lambda_1 I\right)^{-1} XY_{d1}{}^T\right] = \left[W_0 \quad \Delta W_0\right], \end{aligned} \tag{16}$$

where Y_{d1} is an $n \times N$ matrix consisting of N desired feature vectors. Then,

$$\begin{aligned} H_1 &= X^T\left[W_0 \quad \Delta W_0\right] = \left[H_0 \quad X^T \Delta W_0\right] = \left[H_0, \Delta H_0\right], \\ \beta_1 &= \left(\left[H_0, \Delta H_0\right]^T \left[H_0, \Delta H_0\right] + \lambda_2 I\right)^{-1} \left[H_0, \Delta H_0\right]^T T \\ &= \left(\begin{pmatrix} H_0{}^T H_0 & H_0{}^T \Delta H_0 \\ \Delta H_0{}^T H_0 & \Delta H_0{}^T \Delta H_0 \end{pmatrix} + \lambda_2 I\right)^{-1} \\ &\quad \cdot \left[H_0, \Delta H_0\right]^T T \\ &= \begin{pmatrix} H_0{}^T H_0 + \lambda_2 I & H_0{}^T \Delta H_0 \\ \Delta H_0{}^T H_0 & \Delta H_0{}^T \Delta H_0 + \lambda_2 I \end{pmatrix}^{-1} \\ &\quad \cdot \begin{pmatrix} H_0{}^T T \\ \Delta H_0{}^T T \end{pmatrix}. \end{aligned} \tag{17}$$

The Schur complement $P(P = (\Delta H_0{}^T \Delta H_0 + \lambda_2 I) - \Delta H_0{}^T H_0 (H_0{}^T H_0 + \lambda_2 I)^{-1} H_0{}^T \Delta H_0)$ of $H_0{}^T H_0 + \lambda_2 I$ is invertible by choosing the suitable λ_2. Then, using the result on the inversion of 2×2 block matrices [26], we can get

$$\begin{aligned} &\begin{pmatrix} H_0{}^T H_0 + \lambda_2 I & H_0{}^T \Delta H_0 \\ \Delta H_0{}^T H_0 & \Delta H_0{}^T \Delta H_0 + \lambda_2 I \end{pmatrix}^{-1} \\ &= \begin{pmatrix} \left(H_0{}^T H_0 + \lambda_2 I\right)^{-1} + \left(H_0{}^T H_0 + \lambda_2 I\right)^{-1} \left(H_0{}^T \Delta H_0\right) P^{-1} \left(\Delta H_0{}^T H_0\right) \left(H_0{}^T H_0 + \lambda_2 I\right)^{-1} & -\left(H_0{}^T H_0 + \lambda_2 I\right)^{-1} \left(H_0{}^T \Delta H_0\right) P^{-1} \\ -P^{-1} \left(\Delta H_0{}^T H_0\right) \left(H_0{}^T H_0 + \lambda_2 I\right)^{-1} & P^{-1} \end{pmatrix}; \end{aligned} \tag{18}$$

then

$$\begin{aligned}\beta_1 &= \begin{pmatrix} H_0{}^T H_0 + \lambda_2 I & H_0{}^T \Delta H_0 \\ \Delta H_0{}^T H_0 & \Delta H_0{}^T \Delta H_0 + \lambda_2 I \end{pmatrix}^{-1} \begin{pmatrix} H_0{}^T \\ \Delta H_0{}^T \end{pmatrix} T \\ &= \begin{pmatrix} \left(H_0{}^T H_0 + \lambda_2 I\right)^{-1} H_0{}^T T + \left(H_0{}^T H_0 + \lambda_2 I\right)^{-1} \left(H_0{}^T \Delta H_0\right) P^{-1} \left(\Delta H_0{}^T H_0\right) \left(H_0{}^T H_0 + \lambda_2 I\right)^{-1} H_0{}^T T - \left(H_0{}^T H_0 + \lambda_2 I\right)^{-1} \left(H_0{}^T \Delta H_0\right) P^{-1} \Delta H_0{}^T T \\ -P^{-1} \left(\Delta H_0{}^T H_0\right) \left(H_0{}^T H_0 + \lambda_2 I\right)^{-1} H_0{}^T T + P^{-1} \Delta H_0{}^T T \end{pmatrix} \\ &= \begin{pmatrix} \beta_0 + \left(H_0{}^T H_0 + \lambda_2 I\right)^{-1} \left(H_0{}^T \Delta H_0\right) P^{-1} \left(\Delta H_0{}^T H_0\right) \beta_0 - \left(H_0{}^T H_0 + \lambda_2 I\right)^{-1} \left(H_0{}^T \Delta H_0\right) P^{-1} \Delta H_0{}^T T \\ -P^{-1} \left(\Delta H_0{}^T H_0\right) \beta_0 + P^{-1} \Delta H_0{}^T T \end{pmatrix} = \begin{pmatrix} \beta_0 + Q_0 - R_0 \\ U_0 + V_0 \end{pmatrix}. \end{aligned} \tag{19}$$

To save computational cost, $Q_0, R_0, U_0,$ and V_0 in (19) should be computed as the following sequence:

$$\begin{aligned} & P^{-1} \left(\Delta H_0{}^T H_0\right) \longrightarrow P^{-1} \left(\Delta H_0{}^T H_0\right) \beta_0 \left(= U_0\right), \\ & \left(H_0{}^T H_0 + \lambda_2 I\right)^{-1} \left(H_0{}^T \Delta H_0\right) \longrightarrow \left(H_0{}^T H_0 + \lambda_2 I\right)^{-1} \\ & \quad \cdot \left(H_0{}^T \Delta H_0\right) U_0 \left(= Q_0\right), \\ & P^{-1} \Delta H_0{}^T \longrightarrow P^{-1} \Delta H_0{}^T T \left(= V_0\right) \\ & \quad \longrightarrow \left(H_0{}^T H_0 + \lambda_2 I\right)^{-1} \left(H_0{}^T \Delta H_0\right) \\ & \quad \cdot P^{-1} \Delta H_0{}^T T \left(= R_0\right). \end{aligned} \tag{20}$$

Given the number of initial hidden nodes m_0, the maximum number of hidden nodes $m_{\max}$, and the expected output error ε, the OWLM-GHNIL for the SLFNs with the mechanism of growing hidden nodes can be summarized as the following two steps.

Algorithm OWLM-GHNIL

Step 1 (initialization step).

(1) Compute W_0, H_0, β_0 by (13)–(15).

(2) Compute the corresponding output error $E(H_0) = \min \|H_0 \beta_0 - T\|$.

Step 2 (recursively incremental step). Let $k = 0$, and while $m_k < m_{\max}$ and $E(H_k) > \varepsilon$,

(1) $k = k + 1$;

(2) randomly add n (n need not be kept constant) hidden nodes to the existing network; then $W_{k+1}, H_{k+1}, \beta_{k+1}$ can be calculated by (16)–(19).

End

Different from conventional OWLM which needs recalculating the input weight matrix and output weight matrix, whenever the network architecture is changed, the OWLM-GHNIL only needs updating the input weight matrix and output weight matrix incrementally each time; that is why it can reduce the computational complexity significantly. Moreover, the convergence of the OWLM-GHNIL can be guaranteed by the Convergence Theorem in [25].

Now, we begin to analyze computational complexity of the updated work.

The computational complexity, which we consider, expresses the total number of required scalar multiplications. Some matrix computations need not be done repeatedly including the inversion of matrix in (13), since they have been obtained in the process of computing W_k, H_k, β_k. Then it requires lNn, lNn, and $2m_k n^2 + m_k nN + 3m_k nh + 2m_k{}^2 n + 2n^2 N + nNh + n^3$ multiplications for W_{k+1}, H_{k+1}, and β_{k+1}, respectively. Thus, the total computational complexity for the weights W_{k+1} and β_{k+1} is

$$\begin{aligned} C_{\text{OWLM-GHNIL}} = {} & 2lNn + 2m_k n^2 + m_k nN + 3m_k nh \\ & + 2m_k{}^2 n + 2n^2 N + nNh + n^3. \end{aligned} \tag{21}$$

If we compute W_{k+1} and β_{k+1} by (10) and (12) directly, it will cost $C_{\text{OWLM}} = l^3 + 2l^2 N + 2lN(m_k + n) + 2(m_k + n)^2 N + (m_k + n)^3 + (m_k + n)Nh$ multiplications.

Since in most applications m_k and n can be much smaller than the number of training samples N: $m_k, n \ll N$ and h and l are often small number in practical applications, then, with the growth of m_k, n,

$$\frac{C_{\text{OWLM}}}{C_{\text{OWLM-GHNIL}}} \approx \frac{2\left(m_k + n\right)^2}{m_k n + 2n^2}; \tag{22}$$

when $n = (1/2)m_k$,

$$\frac{C_{\text{OWLM}}}{C_{\text{OWLM-GHNIL}}} \approx 4.5; \tag{23}$$

when $n = (1/4)m_k$,

$$\frac{C_{\text{OWLM}}}{C_{\text{OWLM-GHNIL}}} \approx 8.3. \tag{24}$$

It can be seen that the OWLM-GHNIL is much more efficient than the conventional OWLM in such cases.

Similarly, we can get the computational complexity of the EM-ELM and ELM, respectively.

$$\begin{aligned} C_{\text{EM-ELM}} = {} & lNn + 2m_k n^2 + m_k nN + 3m_k nh + 2m_k{}^2 n \\ & + 2n^2 N + nNh + n^3, \end{aligned}$$

$$C_{\mathrm{ELM}} = lN\left(m_k + n\right) + 2\left(m_k + n\right)^2 N + \left(m_k + n\right)^3 + \left(m_k + n\right) Nh. \tag{25}$$

Then, we have

$$C_{\mathrm{OWLM\text{-}GHNIL}} - C_{\mathrm{EM\text{-}ELM}} = lNn,$$
$$C_{\mathrm{OWLM}}\text{-}C_{\mathrm{ELM}} = l^3 + 2l^2 N + lN\left(m_k + n\right). \tag{26}$$

Obviously, the difference on computational complexity between OWLM-GHNIL and EM-ELM is much less than the difference between OWLM and ELM.

4. Simulation Results

In our experiments, all the algorithms are run in such computer environment: (1) operating system: Windows 7 Enterprise; (2) 3.8 GHZ CPU, Intel i5-3570; (3) memory: 8 GB; (4) simulating software: Matlab R2013a.

The performance of the OWLM-GHNIL has been compared with other growing algorithms including the EM-ELM, I-ELM, and the conventional OWLM.

In order to investigate the performance of the proposed OWLM-GHNIL, some benchmark problems are presented in this section.

The OWLM-GHNIL, EM-ELM, and OWLM have first been run to approximate the artificial "SinC" function which is a popular choice to illustrate neural network.

$$y(x) = \begin{cases} \dfrac{\sin x}{x}, & x \neq 0 \\ 1, & x = 0. \end{cases} \tag{27}$$

A training set and testing set with 5000 samples, respectively, are generated from the interval $(-10, 10)$ with random noise distributed in $[-0.2, 0.2]$, while testing data remain noise-free. The performances of each algorithm are shown in Figures 2 and 3. In this case, initial SLFNs are given five hidden nodes and then one new hidden node will be added each step until 30 hidden nodes arrive.

It can be seen from Figure 2 that the OWLM and the OWLM-GHNIL obtain similar lower testing root mean square error (RMSE) than the EM-ELM in most cases. Figure 3 shows the training time comparison of the three methods in SinC case. We can see that, with the growth of hidden nodes, the OWLM-GHNIL spent similar training time with the EM-ELM but much less than the OWLM in the case of the same number of nodes.

In the following, nine real benchmark problems including five regression applications and four classification applications are used for further comparison; all of them are available on the Web. For each case, the training data set and testing data set are randomly generated from its whole data set before each trial of simulation, and average results are obtained over 30 trials for all cases. The features of the benchmark data sets are summarized in Table 1.

The generalization performance comparison between the OWLM-GHNIL and two other popular incremental

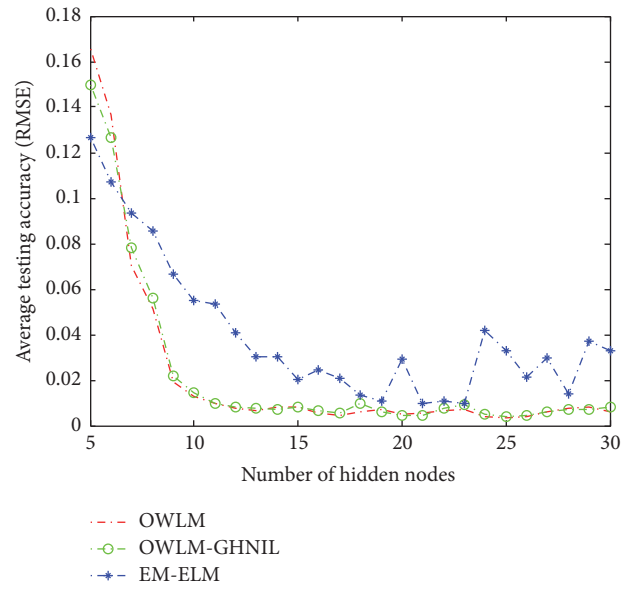


Figure 2: Average testing RMSE of different algorithms in SinC case.

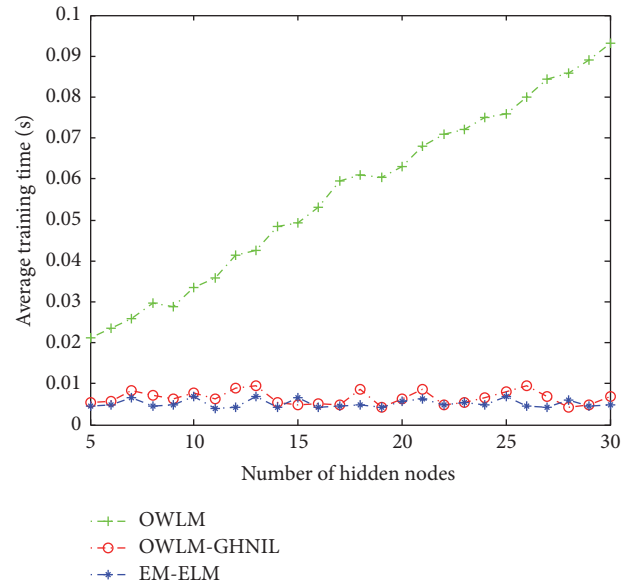


Figure 3: Computational complexity comparison of different algorithms in Sinc case.

ELM-type algorithms, EM-ELM and I-ELM, on regression and classification cases is given in Tables 2 and 3. In the implementation of the EM-ELM and OWLM-GHNIL, initial SLFNs are given 50 hidden nodes and then 25 new hidden nodes will be added each step until 150 hidden nodes arrive. In the case of the I-ELM, the initial SLFNs are given 1 hidden node and then the hidden nodes are added one by one until 150 hidden nodes. As observed from test results of average RMSE and accuracy in Tables 2 and 3, it looks that the OWLM-GHNIL obtained better generalization performance than the EM-ELM and I-ELM. In order to obtain an objective statistical measure, we apply a Student's t-test to each data to check if the differences between the OWLM-GHNIL and

Table 1: Details of the data sets used for regression and classification.

Data sets	Classes	Attributes	Types	# of pieces of training data	# of pieces of testing data
Delta Ailerons	—	6	Regression	3000	4129
Delta Elevators	—	6	Regression	4517	5000
California Housing	—	8	Regression	8000	12,460
Computer activity	—	8	Regression	4000	4192
Bank domains	—	8	Regression	4500	3692
COLL20	20	1024	Classification	1080	360
G50C	2	50	Classification	414	136
USPST(B)	2	256	Classification	1509	498
Satimage	6	36	Classification	3217	3218

Table 2: Comparison of average testing RMSE/accuracy and Student's t-test for the data sets between OWLM-GHNIL and EM-ELM.

Data sets	OWLM-GHNIL		EM-ELM		t-Test
	RMSE/accuracy	Std.	RMSE/accuracy	Std.	p value
Delta Ailerons	0.0583	0.0058	0.1023	0.0081	0.003627
Delta Elevators	0.0896	0.0063	0.1423	0.0154	0.025436
California Housing	0.1841	0.0032	0.2321	0.0045	0.000048
Computer activity	0.0467	0.0021	0.0392	0.0018	0.628794
Bank domains	0.0212	0.0043	0.0611	0.0032	0.007694
COLL20	91.25%	0.0332	86.33%	0.0537	0.000000
G50C	85.31%	0.0553	87.23%	0.0463	0.065632
USPST(B)	90.45%	0.0158	85.21%	0.0547	0.000000
Satimage	88.15%	0.0368	85.37%	0.0446	0.007625

Table 3: Comparison of average testing RMSE/accuracy and Student's t-test for the data sets between OWLM-GHNIL and I-ELM.

Data sets	OWLM-GHNIL		I-ELM		t-Test
	RMSE/accuracy	Std.	RMSE/accuracy	Std.	p-value
Delta Ailerons	0.0583	0.0058	0.1364	0.0223	0.000493
Delta Elevators	0.0896	0.0063	0.2265	0.0196	0.003743
California Housing	0.1841	0.0032	0.4365	0.0223	0.000000
Computer activity	0.0467	0.0021	0.0733	0.0072	0.065475
Bank domains	0.0212	0.0043	0.1128	0.0341	0.000000
COLL20	91.25%	0.0332	78.04%	0.0663	0.000000
G50C	85.31%	0.0553	76.45%	0.0772	0.000482
USPST(B)	90.45%	0.0158	84.33%	0.0472	0.000000
Satimage	88.15%	0.0368	80.33%	0.0682	0.001356

the other two algorithms are statistically significant (p value = 0.05, i.e., confidence of 95%). It was shown in Table 2 that, in four of the regression data sets (Delta Ailerons, Delta Elevators, California Housing, and Bank domains) and three of the classification data sets (COLL20, USPST(B), and Satimage), the t-test gave a significant difference between OWLM-GHNIL and EM-ELM with superior generalization performance of the OWLM-GHNIL, whereas no significant difference was found in the two other data sets (computer activity and G50C). In Table 3, the t-test results show that there was a significant difference between OWLM-GHNIL and I-ELM with superior generalization performance of the OWLM-GHNIL in all data sets except computer activity data set.

5. Conclusion

In this paper, we have developed an efficient method, OWLM-GHNIL; it can grow hidden nodes one by one or group by group in SLFNs. The analysis of computational complexity and simulation results on an artificial problem shows that OWLM-GHNIL can significantly reduce the computational complexity of OWLM. The simulation results on nine real benchmark problems including five regression applications and four classification applications also show that OWLM-GHNIL has better generalization performance than the two other incremental algorithms EM-ELM and I-ELM. t-test gave a significant difference with superior generalization performance of the OWLM-GHNIL further.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by Zhejiang Provincial Natural Science Foundation of China under Grant no. LY18F030003, Foundation of High-Level Talents in Lishui City under Grant no. 2017RC01, Scientific Research Foundation of Zhejiang Provincial Education Department under Grant no. Y201432787, and the National Natural Science Foundation of China under Grant no. 61373057.

References

[1] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, New York, NY, USA, 1995.

[2] G.-B. Huang and H. A. Babri, "Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 9, no. 1, pp. 224–229, 1998.

[3] X.-F. Hu, Z. Zhao, S. Wang, F.-L. Wang, D.-K. He, and S.-K. Wu, "Multi-stage extreme learning machine for fault diagnosis on hydraulic tube tester," *Neural Computing and Applications*, vol. 17, no. 4, pp. 399–403, 2008.

[4] T.-Y. Kwok and D.-Y. Yeung, "Objective functions for training new hidden units in constructive neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 8, no. 5, pp. 1131–1148, 1997.

[5] E. J. Teoh, K. C. Tan, and C. Xiang, "Estimating the number of hidden neurons in a feedforward network using the singular value decomposition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 17, no. 6, pp. 1623–1629, 2006.

[6] X. Luo, J. Deng, W. Wang, J.-H. Wang, and W. Zhao, "A quantized kernel learning algorithm using a minimum kernel risk-sensitive loss criterion and bilateral gradient technique," *Entropy*, vol. 19, no. 7, article no. 365, 2017.

[7] Y. Xu, X. Luo, W. Wang, and W. Zhao, "Efficient DV-HOP localization forwireless cyber-physical social sensing system: A correntropy-based neural network learning scheme," *Sensors*, vol. 17, no. 1, article no. 135, 2017.

[8] S. Haykin, *Neural networks and learning machines*, Pearson, Prentice-Hall, New Jersey, USA, 3rd edition, 2009.

[9] S. Kumar, *Neural Networks*, McGraw-Hill Companies Inc., Columbus, OH, USA, 2006.

[10] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.

[11] G. P. Zhang, "Neural networks for classification: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 30, no. 4, pp. 451–462, 2000.

[12] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley Publishing Company, Boston, Mass, USA, 1991.

[13] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[14] G. Huang, L. Chen, and C. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 17, no. 4, pp. 879–892, 2006.

[15] S.-F. Ding, X.-Z. Xu, and R. Nie, "Extreme learning machine and its applications," *Neural Computing and Applications*, vol. 25, no. 3, pp. 549–556, 2014.

[16] X. Luo, Y. Xu, W. Wang et al., "Towards enhancing stacked extreme learning machine with sparse autoencoder by correntropy," *Journal of The Franklin Institute*, 2017.

[17] M. Anthony and P. L. Bartlett, *Neural Network Learning: Theoretical Foundations*, Cambridge University Press, Cambridge, UK, 1999.

[18] V. N. Vapnik, *Statistical Learning Theory*, Adaptive and Learning Systems for Signal Processing, Communications, and Control, Wiley- Interscience, New York, NY, USA, 1998.

[19] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, vol. 31 of *Stochastic Modelling and Applied Probability*, Springer-Verlag New York, Berlin, Germany, 1996.

[20] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience, New York, NY, USA, 2nd edition, 2001.

[21] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth, Belmont, Mass, USA, 1984.

[22] Z. Man, K. Lee, D. Wang, Z. Cao, and S. Khoo, "An optimal weight learning machine for handwritten digit image recognition," *Signal Processing*, vol. 93, no. 6, pp. 1624–1638, 2013.

[23] X. Luo, J. Deng, J. Liu, W. Wang, X. Ban, and J. Wang, "A quantized kernel least mean square scheme with entropy-guided learning for intelligent data analysis," *China Communications*, vol. 14, no. 7, pp. 127–136, 2017.

[24] W. Zhao, R. Lun, C. Gordon et al., "A human-centered activity tracking system: toward a healthier workplace," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, pp. 343–355, 2017.

[25] G. Feng, G.-B. Huang, Q. Lin, and R. Gay, "Error minimized extreme learning machine with growth of hidden nodes and incremental learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 20, no. 8, pp. 1352–1357, 2009.

[26] G. H. Golub and C. F. van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Md, USA, 3rd edition, 1996.

Hindawi
Scientific Programming
Volume 2018, Article ID 4563040, 10 pages
https://doi.org/10.1155/2018/4563040

Research Article

Robust Matching Pursuit Extreme Learning Machines

Zejian Yuan,[1] Xin Wang,[1] Jiuwen Cao,[2] Haiquan Zhao,[3] and Badong Chen[1]

[1]*Institute of Artificial Intelligence and Robotics, Xian Jiaotong University, Xi'an 710049, China*
[2]*Institute of Information and Control, Hangzhou Dianzi University, Zhejiang 310018, China*
[3]*School of Electrical Engineering, Southwest Jiaotong University, Chengdu, China*

Correspondence should be addressed to Badong Chen; chenbd@mail.xjtu.edu.cn

Received 25 August 2017; Revised 23 November 2017; Accepted 7 December 2017; Published 1 February 2018

Academic Editor: Wenbing Zhao

Extreme learning machine (ELM) is a popular learning algorithm for single hidden layer feedforward networks (SLFNs). It was originally proposed with the inspiration from biological learning and has attracted massive attentions due to its adaptability to various tasks with a fast learning ability and efficient computation cost. As an effective sparse representation method, orthogonal matching pursuit (OMP) method can be embedded into ELM to overcome the singularity problem and improve the stability. Usually OMP recovers a sparse vector by minimizing a least squares (LS) loss, which is efficient for Gaussian distributed data, but may suffer performance deterioration in presence of non-Gaussian data. To address this problem, a robust matching pursuit method based on a novel kernel risk-sensitive loss (in short KRSLMP) is first proposed in this paper. The KRSLMP is then applied to ELM to solve the sparse output weight vector, and the new method named the KRSLMP-ELM is developed for SLFN learning. Experimental results on synthetic and real-world data sets confirm the effectiveness and superiority of the proposed method.

1. Introduction

Extreme learning machine [1] is a kind of single hidden layer feedforward network (SLFN) [2]. In the past decade, ELM became popular and attractive in the machine learning and pattern recognition communities for its fast adaptability and good generalization performance [3]. In general, ELM has the following advantages: (i) It not only has the ability of estimating the unknown mathematical model embedded in a mass of training samples but also possesses parallel schemes to be efficiently implemented in parallel for training and testing; (ii) it uses randomly generated input weights and hidden biases without tuning during the training phase, and therefore, the output weights can be analytically obtained by solving the standard least squares (LS) problem. Thus, extremely fast learning ability and efficient computation cost can be achieved, especially for big data applications. In view of these remarkable superiorities, ELM has been widely applied in many applications, such as face recognition [4], series compensated transmission line protection [5], time series analysis [6], and nonlinear model identification [7].

However, ELM still has several drawbacks. First, ELM encounters the problem of irrelevant variables when handling real-world data sets [8]. Second, choosing a proper hidden nodes number is an open problem for all ELM algorithms. An ELM network with too few hidden nodes may not be accurate for modeling the input data, whereas a network with too many hidden nodes tends to generate an overfitting model [9]. Moreover, when the number of hidden nodes is more than the input data, ELM might have the singularity problem [4]. Third, the original ELM learns the model with an L_2-norm based loss function, which is very vulnerable to noise. It is well known that the L_2-norm can magnify the bad effects of outliers associated with large deviations [10]. The presence of non-Gaussian noises or outliers in the training data may thus lead to an unreliable model with degraded performance.

To overcome the first and second limitations, several methods have been proposed in the regularization framework [9, 11–13]. Furthermore orthogonal matching pursuit (OMP) is a plain and efficient iterative algorithm which chooses an atom in the dictionary with the best correlation to the remaining elements at each iteration [14]. As such, OMP has been

embedded to ELM (OMP-ELM) to overcome the singularity problem and led to more stable solution than the original ELM [15]. Most of the existing methods learn the model with an L_2-norm based loss function, which may perform poorly in the presence of non-Gaussian noises (which exist in many real-world situations) or outliers [16–18]. To combat non-Gaussian noises or outliers and improve the generalization ability, the regularized correntropy criterion is used to replace the L_2-norm based loss function in original ELM model to develop the ELM-RCC [16]. In [19], ELM with L_1-norm based loss function (ORELM) was proposed to achieve robust performance.

The kernel risk-sensitive loss (KRSL) is a nonlinear similarity measure firstly proposed in [20], which can reach a more satisfying robust performance. The KRSL is based on the original structure of risk-sensitive loss and is defined in the reproducing kernel Hilbert space (RKHS) [21, 22]:

$$V(X,Y) = \frac{1}{\lambda}\mathbf{E}\left[\exp\left(\lambda\left(1-\kappa_\sigma(X-Y)\right)\right)\right], \quad (1)$$

where $\mathbf{E}[\cdot]$ denotes the mathematical expectation, $\kappa_\sigma(\cdot)$ is the Gaussian kernel with bandwidth σ, and λ is the risk-sensitive parameter. In this paper, we propose a KRSL based matching pursuit (KRSLMP) method. The KRSLMP is then embedded to ELM to construct a robust and sparse ELM model.

The rest of the paper is structured as follows. In Section 2, we sketch the related work, including similarity measures in kernel space, kernel risk-sensitive loss, ELM model, and orthogonal matching pursuit algorithm. In Section 3, we develop the KRSLMP-ELM. In Section 4, experiments on regression problem with synthetic and real-world data sets are conducted to verify the effectiveness of the proposed algorithm. The sensitivity of the KRSLMP-ELM to free parameters is also analyzed. Finally, conclusion is given in Section 5.

2. Preliminaries and Related Works

For convenience of presentation, the following notations used in this paper are introduced. Vectors and matrices are represented with boldface lowercase letters and boldface capital letters, respectively. For any vector $\mathbf{x}$, we use $x(i)$ to denote its ith entry. The notation $\mathbf{x}|_I$ denotes the subvector of $\mathbf{x} \in \mathbb{R}^n$ with entries indexed by the set $I \subset \Omega = \{1, 2, \ldots, n\}$. The complementary set of I is denoted as $I^c = \Omega - I$.

2.1. Similarity Measures in Kernel Space. Let X and Y be two random variables; the correntropy between X and Y is defined by [17, 23]

$$V(X,Y) = \mathbf{E}\left[\kappa_\sigma(X-Y)\right] = \int \kappa_\sigma(x-y)\, dF_{XY}(x,y), \quad (2)$$

where $F_{XY}(x, y)$ is the joint distribution function of (X, Y). The Gaussian kernel with bandwidth σ is given by

$$\kappa_\sigma(x-y) = \exp\left(-\frac{(x-y)^2}{2\sigma^2}\right). \quad (3)$$

Correntropy $V(X, Y)$ is a local correlation measure in the kernel space $\mathbb{H}$. According to Mercers theorem [24], it can be expressed in terms of the inner product as

$$V(X,Y) = \mathbf{E}\left[\langle \Phi(X), \Phi(Y)\rangle_{\mathbb{H}}\right]. \quad (4)$$

It applies a kernel trick that nonlinearly maps the original space to a higher dimensional feature space. It can be shown that correntropy is directly related to the probability of how similar two random variables are in a neighborhood of the joint space controlled by the kernel bandwidth σ [17, 25, 26].

2.2. Kernel Risk-Sensitive Loss. Similarity measures in kernel space have the ability to extract higher-order statistics of data, which can significantly improve the learning performance in non-Gaussian environments [21]. The optimization problem can be determined by maximizing the correntropy criterion (MCC) or equivalently minimizing the correntropic loss (C-Loss) [27, 28] between the output estimation and the target response. However, highly nonconvex problem may happen in C-Loss performance surface which has steep slopes around the optimal solution but is extremely flat far from the solution. This may lead to slow convergence and poor performance. Choosing a large kernel bandwidth may overcome the above problem. But the robustness will decrease significantly when outliers occur with kernel bandwidth increasing [29]. To achieve a satisfying performance surface, the KRSL was proposed in [20].

The KRSL is defined by

$$\begin{aligned} L_\lambda(X,Y) &= \frac{1}{\lambda}\mathbf{E}\left[\exp\left(\lambda\left(1-\kappa_\sigma(X-Y)\right)\right)\right] \\ &= \frac{1}{\lambda}\int \exp\left(\lambda\left(1-\kappa_\sigma(x-y)\right)\right) dF_{XY}(x,y) \end{aligned} \quad (5)$$

which can also be expressed in a traditional risk-sensitive loss form as [30]

$$L_\lambda(X,Y) = \frac{1}{\lambda}\mathbf{E}\left[\exp\left(\lambda\left(\frac{1}{2}\left\|\Phi(X)-\Phi(Y)\right\|_{\mathbb{H}}^2\right)\right)\right], \quad (6)$$

where λ is the risk-sensitive parameter that controls the shape of performance surface.

In practice, the joint distribution function of X and Y is usually unknown and only a finite number of samples $\{(x_j, y_j)\}_{j=1}^M$ are available. The KRSL can thus be estimated by

$$\widehat{L}_\lambda(X,Y) = \frac{1}{M\lambda}\sum_{j=1}^{M}\exp\left(\lambda\left(1-\kappa_\sigma\left(x_j-y_j\right)\right)\right). \quad (7)$$

As one can see, (6) defines a distance between the vectors $\mathbf{X} = [x_1, x_2, \ldots, x_M]^T$ and $\mathbf{Y} = [y_1, y_2, \ldots, y_M]^T$.

2.3. Extreme Learning Machine. Extreme learning machine (ELM) was proposed by Huang et al. for training single hidden layer feedforward neural networks (SLFNs) [2, 31]. The input weights and biases are initialized randomly in ELM and remain unchanged during training. The network learning thus becomes optimizing the output weights, which can be

formulated as solving a linear equation. Let $\{(\mathbf{x}_j, y_j)\}_{j=1}^{M}$ be given by M training samples, where input $\mathbf{x}_j \in \mathbb{R}^n$ and corresponding desired output $y_j \in \mathbb{R}$; the relationship between $\mathbf{x}_j$ and y_j can be represented under the assumption of the model. The network model of ELM with L hidden neurons can be modeled and expressed as

$$\sum_{i=1}^{L} \beta_i f\left(\mathbf{a}_i \cdot \mathbf{x}_j + b_i\right) = y_j, \quad j = 1, 2, \ldots, M, \tag{8}$$

where L is hidden nodes number, β_i is the weight connecting the ith hidden node and output nodes, f is the activation function (in this work, f is a sigmoid function without explicit mention), $\mathbf{a}_i$ denotes the weight that connects the ith hidden node and input nodes, and b_i represents the randomly chosen bias of the ith hidden node. Equation (7) can be compactly written as a matrix notation

$$\mathbf{y} = \mathbf{H}\boldsymbol{\beta}, \tag{9}$$

where

$$\mathbf{H} = \begin{pmatrix} f\left(\mathbf{a}_1 \cdot \mathbf{x}_1 + b_1\right) & \cdots & f\left(\mathbf{a}_L \cdot \mathbf{x}_1 + b_L\right) \\ \vdots & \ddots & \vdots \\ f\left(\mathbf{a}_1 \cdot \mathbf{x}_M + b_1\right) & \cdots & f\left(\mathbf{a}_L \cdot \mathbf{x}_M + b_L\right) \end{pmatrix} \tag{10}$$

and $\boldsymbol{\beta}$ is the minimal norm least squares solution of (8). The parameter $\boldsymbol{\beta}$ can be obtained by

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{y}, \tag{11}$$

where $\mathbf{H}^{\dagger}$ is the Moore Penrose generalized inverse of the hidden layer output matrix $\mathbf{H}$.

2.4. Orthogonal Matching Pursuit. Matching pursuit method is one of the effective methods for sparse representation [14, 32, 33]. In general, a sparse representation problem can be formulated as

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\rho}, \tag{12}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ $(m < n)$ denotes the measurement matrix, $\mathbf{x}$ is the sparse vector, and $\boldsymbol{\rho} \in \mathbb{R}^m$ represents the noise vector. The main purpose is to recover the sparse vector $\mathbf{x}$ from the observation $\mathbf{y}$ and the measurement matrix $\mathbf{A}$. The OMP uses the L_0-norm constrained least squares model

$$\begin{aligned} \min_{\mathbf{x}} \quad & \left\|\mathbf{y} - \mathbf{A}\mathbf{x}\right\|_2^2 \\ \text{s.t.} \quad & \|\mathbf{x}\|_0 \le K, \end{aligned} \tag{13}$$

where $\|\mathbf{x}\|_0$ counts the number of nonzero coordinates of $\mathbf{x}$.

In the following, we briefly describe the OMP method. First, we initialize the residual $\mathbf{r}_0 = \mathbf{y}$, the index set $\Lambda_0 = \emptyset$, and the iteration $t = 1$. At each iteration, OMP algorithm selects a column of the measurement matrix $\mathbf{A}$ which is most correlated to the residual as

$$\alpha_t = \arg\max_{i=1,\ldots,n} \left|\langle \mathbf{r}_{t-1}, \varphi_i \rangle\right|, \tag{14}$$

where $\mathbf{r}_{t-1}$ denotes the residual in $t-1$th iteration and φ_i is the ith column of $\mathbf{A}$. Then collect α_t to index set Λ

$$\Lambda_t = \Lambda_{t-1} \cup \{\alpha_t\}. \tag{15}$$

We can solve an LS problem to obtain a new estimation $\mathbf{x}_t$ supported in Λ_t:

$$\mathbf{x}_t = \arg\min_{\mathbf{x} \in \mathbb{R}^n, \operatorname{supp}(\mathbf{x}) \subset \Lambda_t} \left\|\mathbf{y} - \mathbf{A}\mathbf{x}\right\|_2, \tag{16}$$

where $\operatorname{supp}(\mathbf{x})$ denotes the support set of $\mathbf{x}$. If the stopping criterion is satisfied, we output $\mathbf{x}_t$ as the estimate of $\mathbf{x}$.

Then one can update the residual

$$\mathbf{r}_t = \mathbf{y} - \mathbf{A}\mathbf{x}_t. \tag{17}$$

From (8) and (11), we can find that ELM has a similar network model for sparse representation problem. Thus, one can take advantage of the OMP algorithm for selecting the best hidden nodes of the ELM network. The OMP estimates the sparse vector by using the L_2-norm based criterion, which performs well with the Gaussian error distribution. However, the presence of non-Gaussian noise may give rise to performance degradation.

3. Kernel Risk-Sensitive Loss Based Matching Pursuit Extreme Learning Machine

To address the aforementioned issue, we propose a robust kernel risk-sensitive loss based orthogonal matching pursuit extreme learning machine algorithm (KRSLMP-ELM) in this section. In the KRSLMP-ELM, we initialize the residual $\mathbf{r}_0$ as $\mathbf{y}$ and the initial index set as $\Lambda_0 = \emptyset$. Then, similar to OMP, a column of H most correlated with the residual is selected and the index set is augmented at each iteration. Then we obtain a new estimation $\boldsymbol{\beta}_t$ by solving the following KRSL minimization problem:

$$\boldsymbol{\beta}_t = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^L, \operatorname{supp}(\boldsymbol{\beta}) \subset \Lambda_t} \phi_{\sigma,\lambda}\left(\mathbf{y} - \mathbf{H}\boldsymbol{\beta}\right) + C\left\|\boldsymbol{\beta}\right\|_2^2. \tag{18}$$

We utilize the half-quadratic (HQ) theory [34] to construct the optimization algorithm. Considering that the measurements may include both large and small noise, we can use HQ optimization to estimate the importance of different samples. The samples severely corrupted will be assigned small weight values in learning procedure to decrease the impact of large noise. Thus, the performance of KRSLMP-ELM can be significantly further improved.

According to the convex optimization theory [35], the dual function for $\phi_{\sigma,\lambda}(x) = (1/\lambda)\exp(\lambda(1 - \exp(-x^2/2\sigma^2)))(0 < \lambda < 1)$ is convex and defined as

$$\psi(s) = \sup_{t \in \mathbb{R}} \left\{-sx^2 + \phi_{\sigma,\lambda}(x)\right\} \tag{19}$$

and then

$$\phi_{\sigma,\lambda}(x) = \inf_{s \in \mathbb{R}} \left\{sx^2 + \psi(s)\right\}, \tag{20}$$

where the infimum is reached at $s = \phi_{\sigma,\lambda}(x)$. We point out here that when the parameter $\lambda > 1$, the KRSLMP-ELM can also work well in our simulations. Substituting (18) for (20), the KRSLMP-ELM objective function can be reformulated as

$$\min_{\substack{\boldsymbol{\beta}\in\mathbb{R}^L,\mathbf{w}\in\mathbb{R}_+^M \\ \mathrm{supp}(\boldsymbol{\beta})\subset\Lambda_k}} J(\boldsymbol{\beta},\mathbf{w}) = \left\| \sqrt{\mathrm{diag}(\mathbf{w})}\,(\mathbf{y}-\mathbf{H}\boldsymbol{\beta}) \right\|_2^2 + C\|\boldsymbol{\beta}\|_2^2 + \sum_{i=1}^{M}\psi(w(i)), \tag{21}$$

where diag($\mathbf{w}$) represents a diagonal matrix with its primary diagonal element $w(i)$ and C is the regularization parameter. Inspired by the HQ theory, (21) can be solved by the following alternate technique:

$$w^{(t+1)}(i) = \frac{1}{\lambda}\exp\left(\lambda\left(1-\kappa_\sigma\left(y(i)-\left(\mathbf{H}\boldsymbol{\beta}^{(t)}\right)(i)\right)\right)\right),$$
$$\boldsymbol{\beta}^{(t+1)} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^L,\mathrm{supp}(\boldsymbol{\beta})\subset\Lambda_k} \left\| \sqrt{\mathrm{diag}\left(\mathbf{w}^{(t+1)}\right)}\,(\mathbf{y}-\mathbf{H}\boldsymbol{\beta}) \right\|_2^2 + C\|\boldsymbol{\beta}\|_2^2, \tag{22}$$

where t denotes the iteration number. In the proposed algorithm, the bandwidth is adaptively chosen during the iteration. In order to make the scheme robust to outliers, we calculate the value of σ as follows.

Denote the training error as $e(i) = \|y(i) - (\mathbf{H}\boldsymbol{\beta})(i)\|_2^2$, $i = 1, 2, \ldots, M$. We can then reorder the error in an ascending order, and we get the reordered as e_σ. Let $k = \lfloor \tau M \rfloor$, where scalar $\tau \in (0, 1]$ and $\lfloor \tau M \rfloor$ outputs the largest integer smaller than τM. We can select $e_\sigma(k)$ as the bandwidth in accordance with the proportion of outlier. Discussions on the detailed experimental results by choosing different bandwidths are given in the experiment section. A solution for the optimization problem in (21) can be derived as follows:

$$\boldsymbol{\beta}^{(t+1)}\big|_{\Lambda_t} = \left(\mathbf{H}^T\mathrm{diag}\left(\mathbf{w}^{(t+1)}\right)\mathbf{H} + \frac{1}{C}\mathbf{I}\right)^{-1}\mathbf{H}^T\mathrm{diag}\left(\mathbf{w}^{(t+1)}\right)\mathbf{y}, \tag{23}$$

where $\boldsymbol{\beta}^{(t+1)}|_{\Lambda_t^c} = 0$ and $\mathbf{I}$ denotes the identity matrix.

Since the importance degree of the measurements is employed to adaptively update the output weight vector in the KRSLMP-ELM, we update the residual

$$\mathbf{r}_t = \sqrt{\mathrm{diag}\left(\mathbf{w}^{(t)}\right)}\left(\mathbf{y}-\mathbf{H}\boldsymbol{\beta}^{(t)}\right). \tag{24}$$

It is noted that the sparsity level K has to be assigned in advance in the KRSLMP-ELM. The sparsity K directly determines the number of the active hidden nodes used in ELM due to the fact that more hidden nodes than necessary are generated. To obtain the best sparsity level K, namely, the best number of hidden nodes used in ELM, we utilize the root mean square error (RMSE) as the criterion

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{M}(y_i-\widehat{y}_i)^2}{M}}, \tag{25}$$

where y_i denotes the target response and $\widehat{y}_i$ the corresponding output estimated by the KRSLMP-ELM.

For different sparsity level K, the corresponding RMSE is first calculated. Then the best $\boldsymbol{\beta}$ coefficients associated with the minimum RMSE value are selected.

The iteration is repeated until achieving the stopping criterion. The KRSLMP-ELM is summarized in Algorithm 1.

4. Experimental Results

To validate the effectiveness of the proposed KRSLMP-ELM algorithm, experiments on two synthetic data sets and seven benchmark data sets are conducted in this section. The performance of the new method is compared to five state-of-the-art algorithms, namely, ELM, RELM, ELM-RCC, OMP-ELM, and ORELM. Sigmoid function $f(x) = 1/(1 + e^{-x})$ is used as the activation function for all methods.

4.1. Synthetic Data Sets. In this subsection, experiments on two synthetic regression data sets for nonlinear function approximation problem are carried out. Descriptions of the two data sets are as follows.

Sinc. The synthetic data set is generated by $y_i = c \cdot \mathrm{Sinc}(x_i) + \rho_i$, where $c = 8$ and

$$\mathrm{Sinc}(x) = \begin{cases} \dfrac{\sin(x)}{x} & x \neq 0 \\ 1 & x = 0 \end{cases} \tag{26}$$

and ρ_i contains two mutually independent noises that are inner noise B_i and outliers noise O_i. Specifically, ρ_i is defined as $\rho_i = (1 - g_i)B_i + g_iO_i$, where g_i is binary distributed with the probability masses $\Pr\{g_i = 1\} = p$ and $\Pr\{g_i = 0\} = 1 - p (0 \leq p \leq 1)$. B_i and O_i are independent of g_i. In this experiment, p is set at 0.1. The outlier O_i is generated by using a zero-mean Gaussian distributed noise with standard deviation 4.0. For the inner noise B_i, two different noises are tested, which are (a) uniform distribution over $[-1.0, 1.0]$ and (b) Sine wave noise $\sin(\alpha)$, with α uniformly distributed over $[0, 2\pi]$. We uniformly generate the input data x_i from $[-10.0, 10.0]$, where 200 data points are used for training and another 200 clean data points which are not contaminated by any noise are used for testing.

Func. This synthetic data set is generated by

$$\mathbf{y}(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1\exp\left\{-\left(\mathbf{x}_1^2 + \mathbf{x}_2^2\right)\right\} + \boldsymbol{\rho}, \tag{27}$$

where $\boldsymbol{\rho}$ is a zero-mean Gaussian distributed noise vector with standard deviation 0.4. The input data vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ are uniformly generated from $[-2.0, 2.0]$. Similar to the previous experiments, 200 data samples are used for training and another 200 data samples without noise are used for testing.

Input: samples $\{\mathbf{x}_i, y_i\}_{i=1}^{M}$
Output: weight vector $\boldsymbol{\beta}$

Parameters setting: number of hidden nodes $\widetilde{L}$, regularization parameter C and sparsity level K.

Initialization: randomly initialize ELM parameters: input weights $\mathbf{a}_i$ and biases b_i $(i = 1, \ldots, L)$ in measurement matrix $\mathbf{H}$.

Set the index set $\Lambda_0 = \emptyset$, the residual $\mathbf{r}_0 = \mathbf{y}$, the iteration counter $t = 0$ and $\mathrm{diag}(\mathbf{w}^0) = \mathbf{I}$.

(1) **for** $t = 1, 2, \ldots, K$ **do**

(2) $t = t + 1$

(3) Find a column of $\mathbf{H}$ most correlated with the residual

$$\alpha_t = \arg\max_{j=1,2,\ldots,L} \left| \left\langle \mathbf{r}_{t-1}, \sqrt{\mathrm{diag}\left(\mathbf{w}^{(t-1)}\right)} \cdot h_j \right\rangle \right|$$

(4) Augment the index set

$$\Lambda_t = \Lambda_{t-1} \cup \{\alpha_t\}$$

(5) Solve the KRSLMP minimization problem by the following iterations

$$w_i^{(t+1)} = \frac{1}{\lambda} \exp\left(\lambda\left(1 - \kappa_\sigma\left(y_i - \left(\mathbf{H}\boldsymbol{\beta}^{(t)}\right)_i\right)\right)\right)$$

$$\boldsymbol{\beta}^{(t+1)} = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^L, \mathrm{supp}(\boldsymbol{\beta}) \subset \Lambda_k} \left\| \sqrt{\mathrm{diag}(\mathbf{w}^{(t+1)})}(\mathbf{y} - \mathbf{H}\boldsymbol{\beta}) \right\|_2^2 + C \left\| \boldsymbol{\beta} \right\|_2^2$$

The solution is denoted as $(\mathbf{w}^{(t)}, \boldsymbol{\beta}^{(t)})$

(6) Update residual $r_t = \sqrt{\mathrm{diag}(\mathbf{w}^{(t)})}(\mathbf{y} - \mathbf{H}\boldsymbol{\beta}^{(t)})$

(7) **end for**

Algorithm 1: KRSLMP-ELM.

Table 1: Parameter settings of four algorithms in function fitting.

	ELM	RELM		OMP-ELM		ORELM		ELM-RCC			KRSLMP-ELM			
	L	L	C	L	K	L	C	L	σ	C	L	K	C	λ
Sinc-Uniform	10	80	10^{-4}	200	10	80	10^{-4}	100	1	10^{-4}	200	50	2×10^{-5}	0.01
Sinc-Sine wave	10	40	2×10^{-5}	200	10	90	10^{-4}	50	1.1	2×10^{-5}	200	50	10^{-5}	0.05
Func	35	35	10^{-6}	200	20	100	10^{-9}	70	3	10^{-4}	200	70	10^{-5}	0.001

Parameters used in the six methods for experiments of the two synthetic data sets are summarized in Table 1, where L, C, K, and λ represent the number of hidden layer nodes, regularization parameter, sparsity level, and risk-sensitive parameter in KRSLMP-ELM. We set $\tau = 0.9$ in Sinc synthetic data set experiment and $\tau = 1$ in Func synthetic data set experiment. For the convenient distinguishment of the proposed method with other methods in Sinc function approximation problem, only the estimation results of the original ELM, ORELM, ELM-RCC, and KRSLMP-ELM are illustrated in Figure 1. In Figure 2, we plot the squared training errors obtained by the KRSLMP-ELM, ELM-RCC, ORELM, and the original ELM, respectively. As shown in these figures, the KRSLMP-ELM wins the best approximation performance. The testing RMSEs of six algorithms are presented in Table 2. It is indicated that the KRSLMP-ELM is more robust than the other five methods.

Further, we perform another experiment to compare the performance of KRSLMP-ELM to that of the original ELM with different outliers. We consider the Sinc function approximation problem and set the inner noise as a zero-mean Gaussian distributed noise with standard deviation 0.1, and the outliers noise is zero-mean Gaussian with standard deviation ranging between 0.1 and 10. We run 100 trials for different outliers noises and show the RMSE results in Figure 3. One can see that the original ELM's performance degrades severely when the outliers get enhanced while the KRSLMP-ELM's performance is much less influenced by outliers.

4.2. Benchmark Data Sets. In this subsection, seven benchmark regression data sets from UCI machine learning repository [36] are tested to support the superiority of the proposed method. Specifications of the data sets are shown detailedly in Table 3. It should be pointed out that the training and testing data samples are randomly chosen in each data set and all the features are normalized into $[0, 1]$. The parameters of each method are all chosen by the fivefold cross-validation and are given in Table 4. For all algorithms, 100 independent trials are conducted and the average results are reported. The training and testing RMSEs and their standard deviation of all algorithms are listed in Table 5. As highlighted in boldface, the ELM-KRSLMP achieves the best performance in most regression data sets.

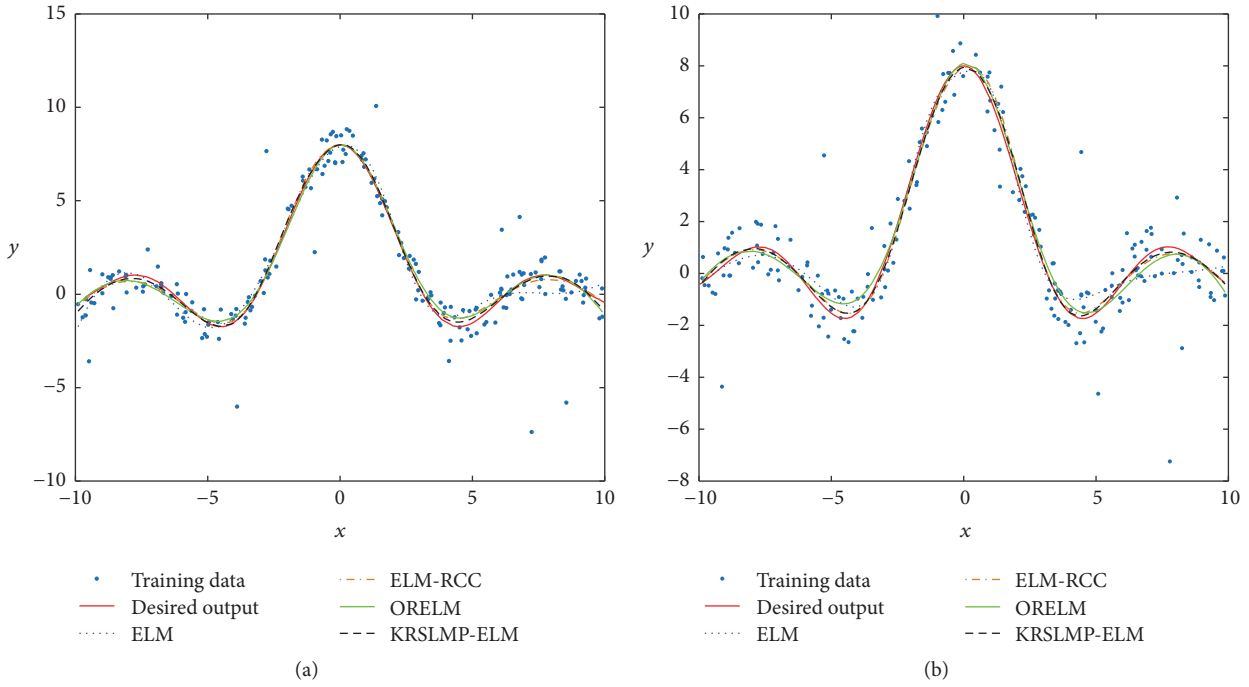


Figure 1: Sinc function regression results with different inner noises: (a) Uniform; (b) Sine wave.

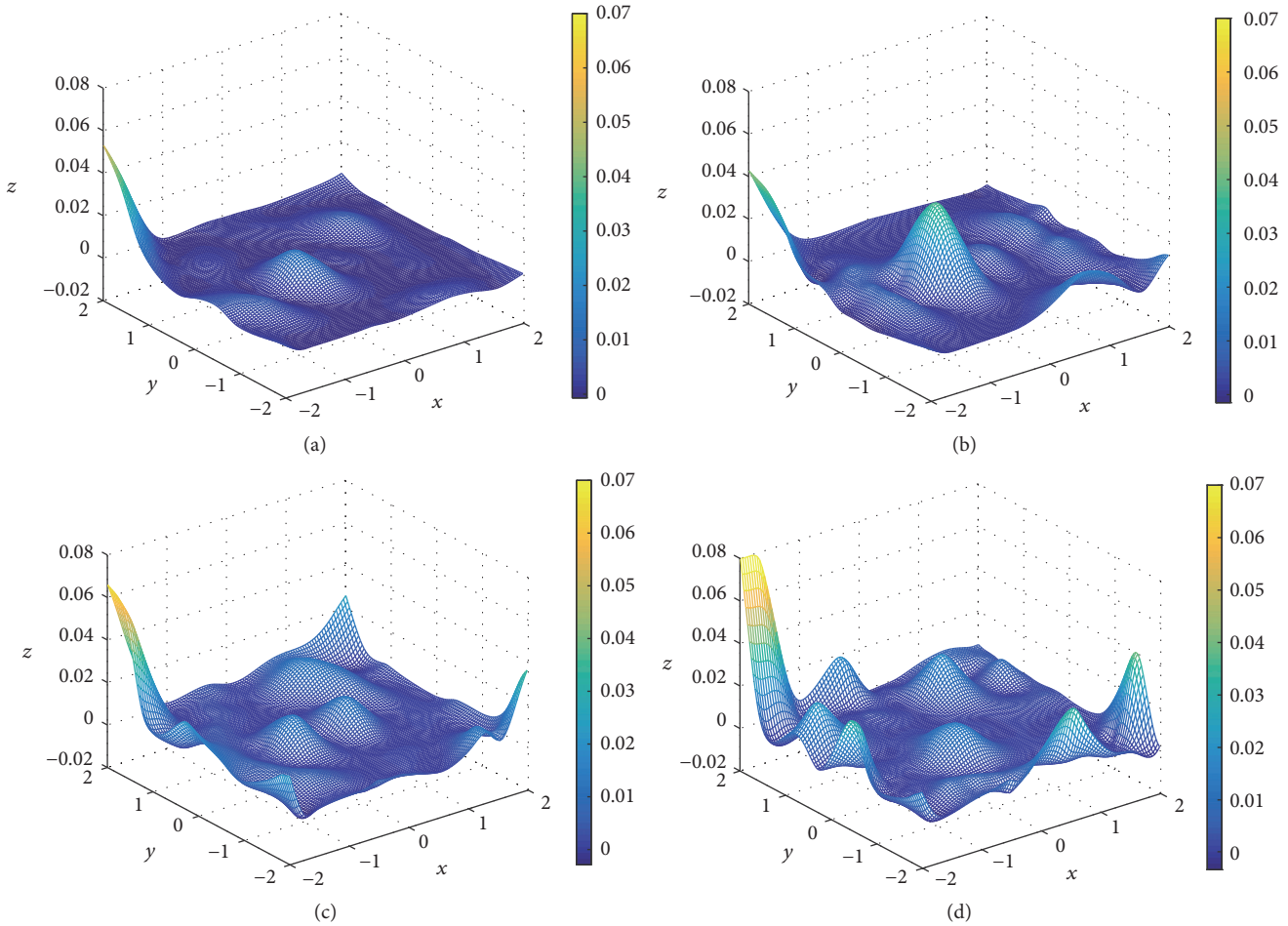


Figure 2: Squared training errors of Func function regression: (a) KRSLMP-ELM; (b) ELM-RCC; (C) ORELM; (D) original ELM.

Table 2: Testing RMSEs of six methods.

	ELM	RELM	OMP-ELM	ORELM	ELM-RCC	KRSLMP-ELM
Sinc-Uniform	0.4737	0.2871	0.3038	0.2369	0.1948	**0.1485**
Sinc-Sine wave	0.4406	0.2935	0.2711	0.2798	0.2155	**0.1562**
Func	0.0652	0.0638	0.0637	0.0591	0.0582	**0.0434**

Table 3: Specification of the data sets.

Data sets	Features	Observations	
		Training	Testing
Servo	5	83	83
Auto MPG	7	192	200
Body fat	14	126	126
Concrete	9	515	515
Housing	14	253	253
Yacht	6	154	154
Airfoil	5	751	751

Table 4: Parameter settings of six methods.

	ELM	RELM		OMP-ELM		ORELM		ELM-RCC			KRSLMP-ELM			
	L	L	C	L	K	L	C	L	σ	C	L	K	C	λ
Servo	25	90	10^{-5}	100	20	120	10^{-6}	65	0.8	10^{-4}	200	40	5×10^{-5}	1.5
Auto MPG	20	40	10^{-4}	50	15	100	10^{-4}	100	0.3	10^{-2}	200	80	10^{-3}	0.8
Body fat	20	100	10^{-2}	50	15	100	10^{-3}	160	0.1	10^{-1}	100	25	10^{-2}	1.1
Concrete	120	185	2×10^{-4}	200	80	200	10^{-7}	200	0.6	5×10^{-6}	500	140	10^{-5}	0.6
Housing	40	180	2×10^{-4}	200	30	200	10^{-5}	200	0.8	10^{-3}	500	150	10^{-3}	0.5
Yacht	90	185	2×10^{-5}	200	60	200	10^{-9}	195	0.4	10^{-7}	500	145	10^{-8}	0.3
Airfoil	130	200	2×10^{-4}	200	85	180	10^{-9}	150	0.4	10^{-7}	500	140	10^{-8}	1.0

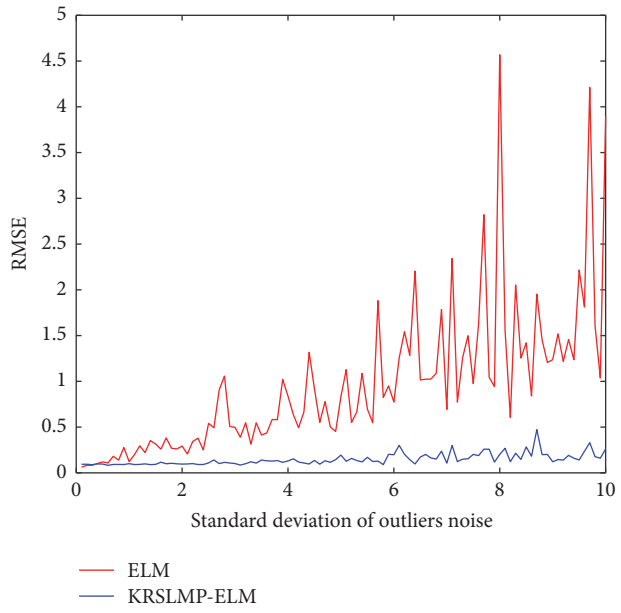


Figure 3: Sinc function regression results with different outliers noises.

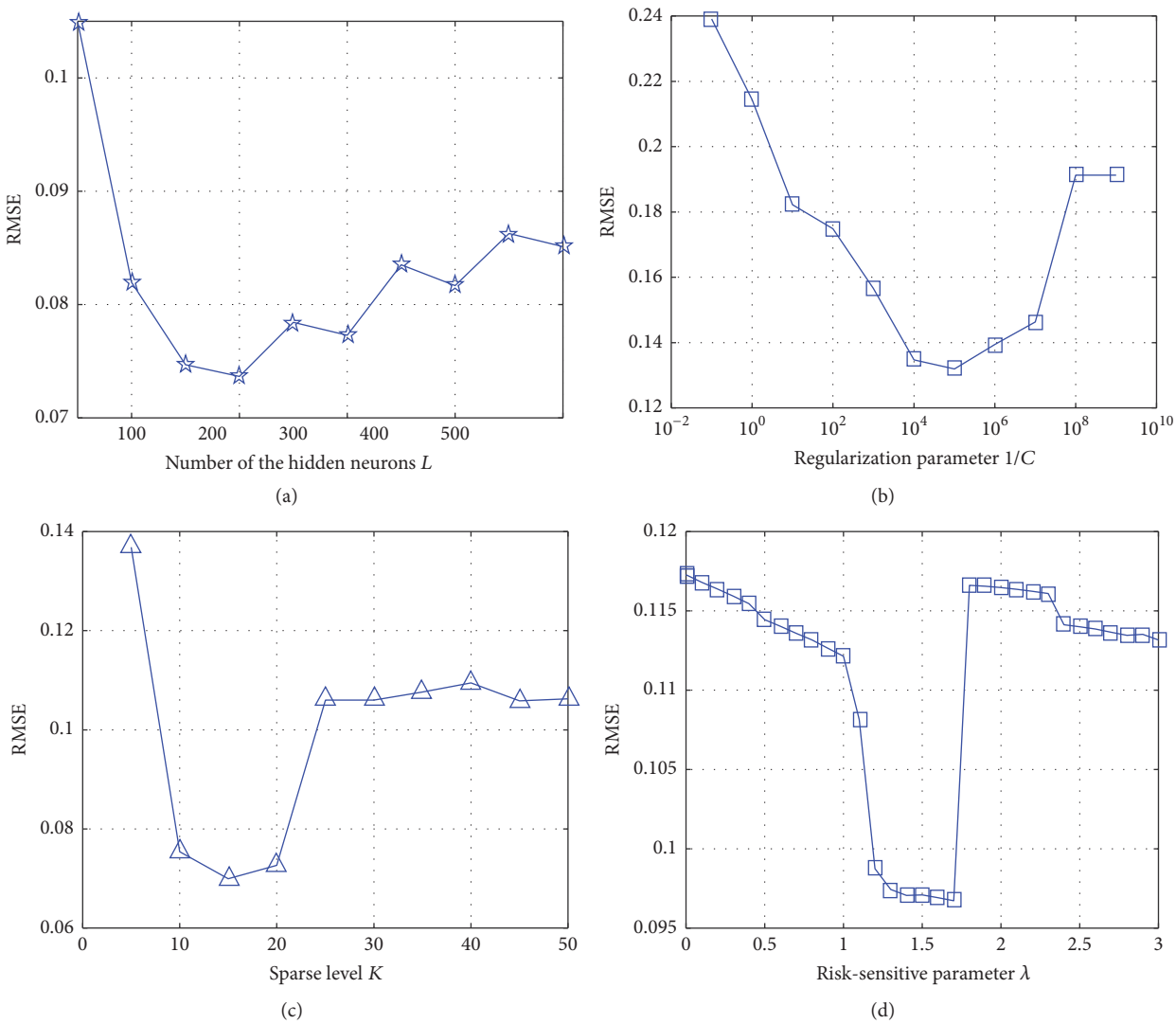


Figure 4: Regression results with different parameters: (a) L; (b) $1/C$; (c) K; (d) λ.

4.3. Sensitivity of Parameters. We analyze the sensitivity of the parameters L, K, C, and λ of KRSLMP-ELM in this subsection. For illustration, we use the regression results obtained by the Servo data set as an example. For each parameter, its sensitivity is tested by fixing the remaining parameters as the ones used in Table 4. Then, the testing RMSEs are recorded as criteria for performance comparison. The results of the regression performance are demonstrated in Figure 4.

5. Conclusion

In this paper, a robust matching pursuit based ELM algorithm, called the kernel risk-sensitive loss based matching pursuit extreme learning machine (KRSLMP-ELM), has been developed. Kernel risk-sensitive loss (KRSL) is a nonlinear similarity measure defined in kernel space, and it can achieve better performance than the conventional MSE criterion when dealing with non-Gaussian and nonlinear problems. Incorporating the KRSL into the existing orthogonal matching pursuit algorithm, we developed an improved KRSLMP-ELM algorithm, which is more robust than the OMP-ELM method. Comparisons with several existing state-of-the-art algorithms have also been provided to validate the superiority of the proposed KRSLMP-ELM algorithm.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation-Shenzhen Joint Research Program (no.

Table 5: Training and testing RMSEs for different data sets.

Data sets	ELM		RELM		OMP-ELM		ORELM		ELM-RCC		KRSLMP-ELM	
	Training RMSE	Testing RMSE	Training RMSE	Testing RMSE	Training RMSE	Testing RMSE	Training RMSE	Testing RMSE	Training RMSE	Testing RMSE	Training RMSE	Testing RMSE
Servo	0.0745 ± 0.0133	0.1189 ± 0.0228	0.0583 ± 0.0105	0.1044 ± 0.0189	0.0727 ± 0.0123	0.1134 ± 0.0181	0.0849 ± 0.0174	0.1036 ± 0.0211	0.0749 ± 0.0117	0.1034 ± 0.0171	0.0841 ± 0.0115	**0.1017 ± 0.0176**
Auto MPG	0.0689 ± 0.0049	0.0785 ± 0.0053	0.0627 ± 0.0044	0.0765 ± 0.0043	0.0676 ± 0.0044	0.0777 ± 0.0045	0.0705 ± 0.0047	0.0764 ± 0.0052	0.0683 ± 0.0042	0.0757 ± 0.0045	0.0632 ± 0.0043	**0.0749 ± 0.0045**
Body fat	0.0237 ± 0.0073	0.0340 ± 0.0060	0.0196 ± 0.0076	0.0278 ± 0.0062	0.0218 ± 0.0076	0.0313 ± 0.0059	0.0253 ± 0.0118	**0.0233 ± 0.0119**	0.0240 ± 0.0093	0.0236 ± 0.0087	0.0252 ± 0.0097	0.0239 ± 0.0095
Concrete	0.0615 ± 0.0025	0.1001 ± 0.0125	0.0732 ± 0.0022	0.0925 ± 0.0041	0.0654 ± 0.0026	0.0981 ± 0.0101	0.0668 ± 0.0026	0.0929 ± 0.0069	0.0557 ± 0.0018	0.0882 ± 0.0077	0.0542 ± 0.0018	**0.0864 ± 0.0067**
Housing	0.0736 ± 0.0053	0.0990 ± 0.0094	0.0443 ± 0.0041	0.0897 ± 0.0136	0.0644 ± 0.0048	0.0935 ± 0.0108	0.0576 ± 0.0061	0.0899 ± 0.0160	0.0453 ± 0.0039	0.0874 ± 0.0129	0.0503 ± 0.0040	**0.0849 ± 0.0114**
Yacht	0.0041 ± 0.0004	0.0583 ± 0.1374	0.0300 ± 0.0002	0.0483 ± 0.0064	0.0154 ± 0.0014	0.0437 ± 0.0134	0.0189 ± 0.0013	0.0373 ± 0.0073	0.0126 ± 0.0008	0.0320 ± 0.0063	0.0060 ± 0.0005	**0.0250 ± 0.0099**
Airfoil	0.0664 ± 0.0020	0.0967 ± 0.0130	0.0925 ± 0.0021	0.0991 ± 0.0023	0.0731 ± 0.0024	0.0968 ± 0.0105	0.0806 ± 0.0023	0.0951 ± 0.0052	0.0740 ± 0.0021	0.0907 ± 0.0061	0.0635 ± 0.0021	**0.0882 ± 0.0068**

U1613219) and National Natural Science Foundation of China (no. 91648208 and no. 61372152).

References

[1] G. Huang, "An insight into extreme learning machines: random neurons, random features and kernels," *Cognitive Computation*, 2014.

[2] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[3] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 985–990, July 2004.

[4] W. Zong and G.-B. Huang, "Face recognition based on extreme learning machine," *Neurocomputing*, vol. 74, no. 16, pp. 2541–2551, 2011.

[5] V. Malathi, N. S. Marimuthu, S. Baskar, and K. Ramar, "Application of extreme learning machine for series compensated transmission line protection," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 5, pp. 880–887, 2011.

[6] R. Singh and S. Balasundaram, "Application of extreme learning machine method for time series analysis," in *Proceedings of the Rampal Singh and S Balasundaram. Application of extreme learning machine method for time series analysis. International Journal of Intelligent Technology*, vol. 2, pp. 256–262, 2007.

[7] J. Deng, K. Li, and G. W. Irwin, "Fast automatic two-stage nonlinear model identification based on the extreme learning machine," *Neurocomputing*, vol. 74, no. 16, pp. 2422–2429, 2011.

[8] Y. Miche, M. van Heeswijk, P. Bas, O. Simula, and A. Lendasse, "TROP-ELM: a double-regularized ELM using LARS and Tikhonov regularization," *Neurocomputing*, vol. 74, no. 16, pp. 2413–2421, 2011.

[9] W. Y. Deng, Q. H. Zheng, and L. Chen, "Regularized extreme learning machine," in *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM '09)*, pp. 389–395, April 2009.

[10] H. Wang, "Block principal component analysis with L1-norm for image analysis," *Pattern Recognition Letters*, vol. 33, no. 5, pp. 537–542, 2012.

[11] J. M. Martínez-Martínez, P. Escandell-Montero, E. Soria-Olivas, J. D. Martín-Guerrero, R. Magdalena-Benedito, and J. Gómez-Sanchis, "Regularized extreme learning machine for regression problems," *Neurocomputing*, vol. 74, no. 17, pp. 3716–3721, 2011.

[12] L.-C. Shi and B.-L. Lu, "EEG-based vigilance estimation using extreme learning machines," *Neurocomputing*, vol. 102, pp. 135–143, 2013.

[13] Y. Miche, P. Bas, C. Jutten, O. Simula, and A. Lendasse, "A methodology for building regression models using extreme learning machine: OP-ELM," in *Proceedings of the 16th European Symposium on Artificial Neural Networks—Advances in Computational Intelligence and Learning (ESANN '08)*, pp. 247–252, April 2008.

[14] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[15] O. F. Alcin, A. Sengur, J. Qian, and M. C. Ince, "OMP-ELM: Orthogonal matching pursuit-based extreme learning machine

for regression," *Journal of Intelligent Systems*, vol. 24, no. 1, pp. 135–143, 2015.

[16] H.-J. Xing and X.-M. Wang, "Training extreme learning machine via regularized correntropy criterion," *Neural Computing and Applications*, vol. 23, no. 7-8, pp. 1977–1986, 2013.

[17] W. Liu, P. P. Pokharel, and J. C. Principe, "Correntropy: properties and applications in non-Gaussian signal processing," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5286–5298, 2007.

[18] Y. Feng, X. Huang, L. Shi, Y. Yang, and J. A. Suykens, "Learning with the maximum correntropy criterion induced losses for regression," *Journal of Machine Learning Research (JMLR)*, vol. 16, pp. 993–1034, 2015.

[19] K. Zhang and M. Luo, "Outlier-robust extreme learning machine for regression problems," *Neurocomputing*, vol. 151, no. 3, pp. 1519–1527, 2015.

[20] B. Chen, L. Xing, B. Xu, H. Zhao, N. Zheng, and J. C. Principe, "Kernel risk-sensitive loss: definition, properties and application to robust adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 65, no. 11, pp. 2888–2901, 2017.

[21] J. C. Principe, *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*, Information Science and Statistics, Springer, New York, NY, USA, 2010.

[22] X. Luo, J. Deng, J. Liu, W. Wang, X. Ban, and J. Wang, "A quantized kernel least mean square scheme with entropy-guided learning for intelligent data analysis," *China Communications*, vol. 14, no. 7, pp. 127–136, 2017.

[23] B. Chen and J. C. Príncipe, "Maximum correntropy estimation is a smoothed MAP estimation," *IEEE Signal Processing Letters*, vol. 19, no. 8, pp. 491–494, 2012.

[24] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT press, Cambridge, Massachusetts, USA, 2002.

[25] W. Ma, J. Duan, B. Chen, G. Gui, and W. Man, "Recursive generalized maximum correntropy criterion algorithm with sparse penalty constraints for system identification," *Asian Journal of Control*, vol. 19, no. 3, pp. 1164–1172, 2017.

[26] I. Santamaría, P. P. Pokharel, and J. C. Principe, "Generalized correlation function: definition, properties, and application to blind equalization," *IEEE Transactions on Signal Processing*, vol. 54, no. 6 I, pp. 2187–2197, 2006.

[27] B. Chen, Y. Zhu, J. Hu, and J. Principe, "System Parameter Identification: Information Criteria and Algorithms," *System Parameter Identification: Information Criteria and Algorithms*, pp. 1–249, 2013.

[28] B. Chen, J. Wang, H. Zhao, N. Zheng, and J. C. Principe, "Convergence of a fixed-point algorithm under maximum correntropy criterion," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1723–1727, 2015.

[29] X. Luo, J. Deng, W. Wang, J.-H. Wang, and W. Zhao, "A quantized kernel learning algorithm using a minimum kernel risk-sensitive loss criterion and bilateral gradient technique," *Entropy*, vol. 19, no. 7, article no. 365, 2017.

[30] R. K. Boel, M. R. James, and I. R. Petersen, "Robustness and risk-sensitive filtering," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 47, no. 3, pp. 451–461, 2002.

[31] X. Luo, Y. Xu, W. Wang et al., "Towards enhancing stacked extreme learning machine with sparse autoencoder by correntropy," *Journal of The Franklin Institute*, 2017.

[32] M. A. Davenport and M. B. Wakin, "Analysis of orthogonal matching pursuit using the restricted isometry property," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 56, no. 9, pp. 4395–4401, 2010.

[33] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 40–44, Pacific Grove, Calif, USA, November 1993.

[34] M. Nikolova and M. K. Ng, "Analysis of half-quadratic minimization methods for signal and image recovery," *SIAM Journal on Scientific Computing*, vol. 27, no. 3, pp. 937–966, 2005.

[35] R. T. Rockafellar, *Convex Analysis*, Princeton Mathematical Series, No. 28, Princeton University Press, Princeton, NJ, USA, 1970.

[36] A. Frank, *Uci machine learning repository*, 2010, http://archive.ics.uci.edu/ml.

Hindawi
Scientific Programming
Volume 2018, Article ID 1327214, 13 pages
https://doi.org/10.1155/2018/1327214

Review Article

Big Data Management for Cloud-Enabled Geological Information Services

Yueqin Zhu[1,2], **Yongjie Tan**[1,2], **Xiong Luo**[3,4] and **Zhijie He**[3,4]

[1] *Development and Research Center, China Geological Survey, Beijing 100037, China*
[2] *Key Laboratory of Geological Information Technology, Ministry of Land and Resources, Beijing 100037, China*
[3] *School of Computer and Communication Engineering, University of Science and Technology Beijing (USTB), Beijing 100083, China*
[4] *Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing 100083, China*

Correspondence should be addressed to Yueqin Zhu; yueqin_zhu@126.com and Xiong Luo; xluo@ustb.edu.cn

Received 20 October 2017; Revised 10 December 2017; Accepted 31 December 2017; Published 29 January 2018

Academic Editor: Anfeng Liu

Cloud computing as a powerful technology of performing massive-scale and complex computing plays an important role in implementing geological information services. In the era of big data, data are being collected at an unprecedented scale. Therefore, to ensure successful data processing and analysis in cloud-enabled geological information services (CEGIS), we must address the challenging and time-demanding task of big data processing. This review starts by elaborating the system architecture and the requirements for big data management. This is followed by the analysis of the application requirements and technical challenges of big data management for CEGIS in China. This review also presents the application development opportunities and technical trends of big data management in CEGIS, including collection and preprocessing, storage and management, analysis and mining, parallel computing based cloud platform, and technology applications.

1. Introduction

In the era of big data, the data-driven modeling method enables us to exploit the potential of massive amount of geological data easily [1–3]. In particular, by mining the data scientifically, one can offer new services that bring higher values to customers. Furthermore, it is now possible to implement the transition from digital geology to intelligent geology by integrating multiple systems in geological research through the use of big data and other technologies [4].

The application of geological cloud makes it possible to fully utilize structured and unstructured data, including geology, minerals, geophysics, geochemistry, remote sensing, terrain, topography, vegetation, architecture, hydrology, disasters, and other digitally geological data distributed in every place on the surface of the earth [4, 5]. Moreover, the geological cloud will enable the integration of data collection, resource integration, data transmission, information extraction, and knowledge mining, which will pave the way for the transition from data to information, from information to knowledge, and from knowledge to wisdom. In addition, it provides data analysis, mining, organization, and management services for the scientific management of land resources, prospecting breakthrough strategic action and social services, while conducting multilevel, multiangle, and multiobjective demonstration applications on geological data for government decision-making, scientific research, and public services [5].

Big data technologies are bringing unprecedented opportunities and challenges to various application areas, especially to geological information processing [2, 6, 7]. Under these circumstances, there are some advancements achieved in the development of this area [8, 9]. Furthermore, from various disciplines of science and engineering, there has been a growing interest in this research field related to geological data generated in the geological information services (GIS). We analyze the number of those documents indexed in "Web of Science" [10]. In Figures 1 and 2, we can easily find that, in the past ten years, the number of those documents in which "geological data" is in the "Title" and in the "Topic" are both increasing, respectively. Hence, the analysis for geological

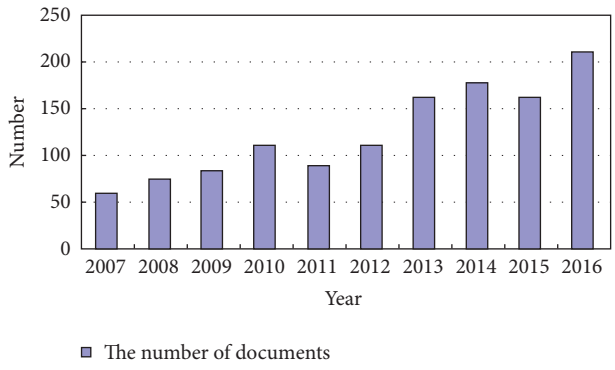


Figure 1: The trend of the number of documents in which "geological data" is in the "Title" from 2007 to 2016.

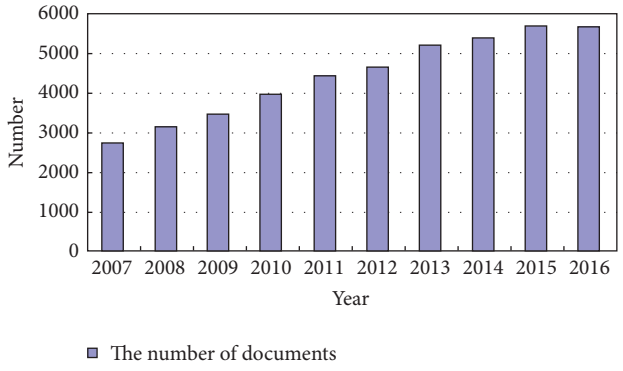


Figure 2: The trend of the number of documents in which "geological data" is in the "Topic" from 2007 to 2016.

data in GIS is an interesting and important research topic currently.

Considering the development status of cloud-enabled geological information services (CEGIS) and the application requirements of big data management analysis, this article describes the significant impact and revolution on GIS brought by the advancement of big data technologies. Furthermore, this article outlines the future application development and technology development trend of big data management analysis in CEGIS.

The remainder of this article is organized as follows. In Section 2, we provide a review on CEGIS, with an emphasis on the descriptions for the system architecture and those requirements from big data management. Then, the challenges for big data management in CEGIS are presented in Section 3. Furthermore, the key technologies and trends on big data management in CEGIS are analyzed in Section 4. Finally, conclusion is drawn in Section 5.

2. Review on Cloud-Enabled Geological Information Services

The construction of geological cloud differs from the current big data analysis based on Internet and Internet of Things (IoT). Having a deep understanding of data characteristics is necessary to collect, process, analyze, and interpret data in different fields, because the nature and types of data vary in different fields and in different problems. Geology is a data intensive science and geological data are characterized with multisource heterogeneity, spatiotemporal variation, correlation, uncertainty, fuzziness, and nonlinearity. Therefore, the geological cloud has a certain degree of confidentiality and it is highly domain-specific; meanwhile, it is developed on the basis of a large amount of geological data accumulated over a long period of time [5, 11]. There are many real-time data generated from some issues like geological disasters and geological environment. The geological cloud includes core basic data, which can be divided into three parts: existing structured database, some unstructured data, and public application data. Therefore, it is important to take good advantage of the existing traditional structured data, use the big data technologies to deal with the relevant unstructured data, and also consider the peripheral public data.

Geological big data are multidimensional, and they consist of both structured and unstructured data [12]. The technical methods of big data analysis differ greatly from those of professional databases. Long-term geological survey, geological study, and years of geological information accumulation have formed a rich and professional database, which is an important fundamental assurance for land and resources science management, geological survey, and geological information public service [13]. This "professional cloud" objectively requires the technology research and development, such as construction of professional local area network, data sharing platform, and geological big data visualization services. Hence, the construction of geological cloud is closely related to land resource management, deployment decision, and the application demand of public service. The key technologies of research and development include the following: unstructured data extraction and mining analysis, structured and unstructured data mixed storage and management, big data sharing platform, data transmission, and visualization [11].

Generally, the construction of geological cloud is a long-term systematic project. This means that it is required to follow the basic principles of "standing on the reality, focusing on the future" and "focusing on the long-term and overall situation, embarking on the current and local situation," in order to achieve the analysis and application of geological cloud public data and core data gradually in accordance with the technical route of big data analysis; thus the construction of geological cloud will be implemented eventually. For the earth, the land and resources management should cover many respects, including human behavior, climate change, development and utilization of various resources, natural disasters, environmental pollution, and the ecosystem cycle. Then, the introduction of big data technologies can integrate this type of resource information to provide the ability of uniformly dealing with the problems related to the entire earth information resources, which has a significant effect on the strategic planning of land and resources management [3].

The geological cloud is an important part in the science system for geological data research. The ultimate goal for developing geological cloud is to better describe and understand the complex earth system and geological framework,

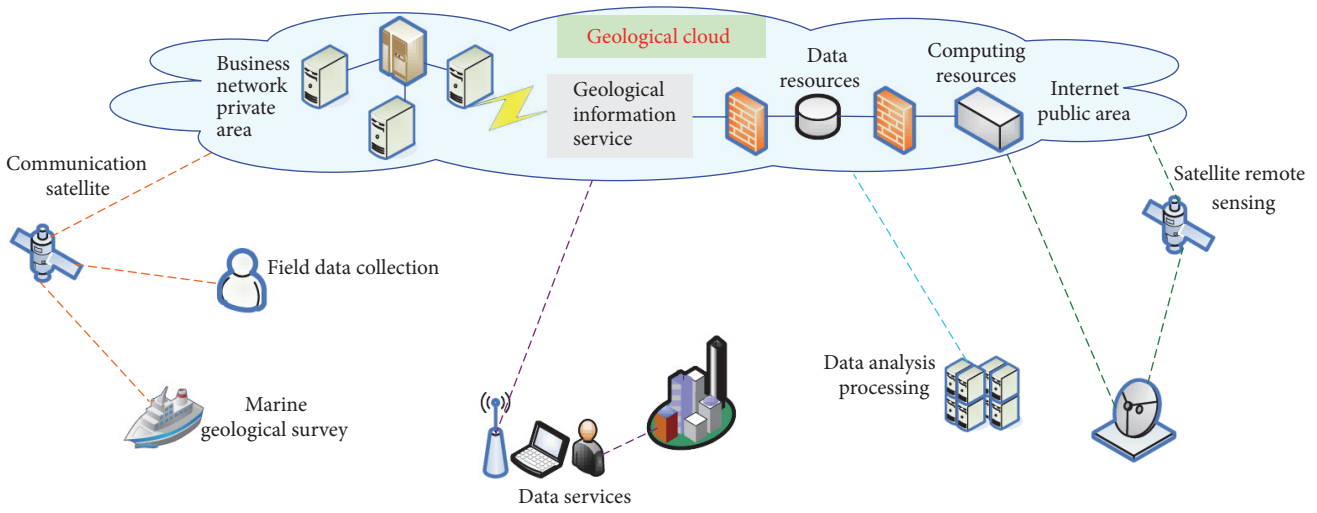


Figure 3: The system architecture of geological cloud.

provide the scientific basis for the description of the land surface and the biodiversity characteristics of the earth, and improve the ability to deal with complex social problems.

2.1. System Architecture. Because the business service functions of each country are different, the system architecture of the geological cloud would vary. In the following, we present a system architecture in Figure 3 [14], using China as an example.

The geological cloud combines the geological survey Intranet and the geological survey Extranet. It enables the sharing and management of computing resources, storage resources, network resources, software resources, and geological data resources [15].

Geological cloud can be summarized with the following characteristics [14].

(i) "*One Platform: The Geological Cloud Management Platform.*" It uniformly manages computing resources, storage resources, network resources, software resources, and geological data resources.

(ii) "*Two Networks: The Geological Survey Intranet and the Geological Survey Extranet.*" Here, the Intranet is constructed by creating a network that is physically isolated from the Internet. The Intranet is developed on the basis of the existing geological survey network and each node is linked through a dedicated line or bare fiber. All of the internal business management systems, software systems, and data are deployed on the Internet, providing services to 28 local units and those users of more than 350 geological survey projects. Facilitated by the public geological survey network, the geological survey business management system, geological data information service system, and public geological data can be deployed on the Extranet accessed by the general public. The communication between the Intranet and the Extranet, including data exchange and audit, can be carried out by single-directional light gate.

(iii) "*One Main Node and Three Domain-Specific Nodes.*" One main node is constructed in China Geological Survey Development Research Center. In addition, three domain-specific nodes—namely, marine node, geological environment node, and aviation geophysical exploration and remote sensing node—are constructed, respectively. Each node is configured with the corresponding servers, storage equipment, network equipment, management platform, large-scale specialized data processing software system and various customized applications. Each node would store huge amounts of geological data and conform to current data security standards. The master node and the domain-specific nodes are linked via optical fibers. The master node will consist of 200 computing nodes with 3 PB storage capacity and will be equipped with some geological data processing software system. The master node will be hosted in a medium-sized supercomputing center and it will provide support for the three-dimensional seismic exploration data processing and other large-scale computing. The three domain-specific nodes are to maintain their scale in the near future to facilitate reasonable scheduling and efficient utilization of information resources and data resources.

On the Extranet, it deploys a system for geological survey business management and auxiliary decision-making. The system provides a real-time tracking and management function for geological survey projects and various resources.

Main users of the geological cloud include institutional users, geological survey project users, and the general public users. The institutional users can store the current geological database and newly collected data in the geological cloud through the geological survey business network and can

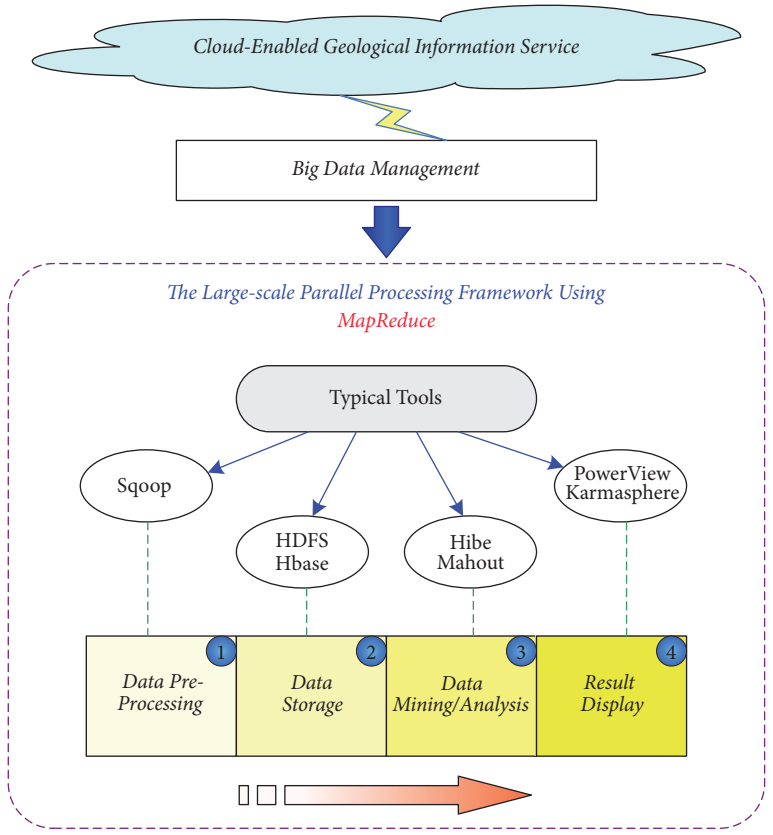


Figure 4: Schematic diagram of big data analysis.

obtain the geological data of other institutions from the cloud as needed. The geological survey project users can access the cloud geological background data through 4G or satellite lines and can collect data through data the collection system.

2.2. Requirement from Big Data Management. The construction of geological cloud must meet customer demand. Big data technologies are then used as the means to implement the geological cloud.

The types and quantity of geological data have been continuously growing over the years. Geological data include all kinds of electronic documents, structured, semistructured, and unstructured data, such as various databases (map database, spatial database, and attribute database), pictures, tables, video, and audio. Generally speaking, those important data may be buried in the massive data without the guidance for requirements. Hence, the first step is to understand the user requirements and then gain the capability of large-scale data processing. This is followed by data mining, algorithm, and analysis, which will ultimately generate value. Big data technologies in the field of geography must meet different needs from people at different levels, including the public demand of the geologic data services and professional data demand for geological research institutions, as well as related enterprises and government departments [16].

On the basis of big data analysis technologies, a complete data link is formed connecting data, information, knowledge, and service, through the use of advanced cloud computing system, IoT, and big data processing flow. It is shown in Figure 4 [5].

3. Challenges for Big Data Management in Cloud-Enabled Geological Information Services

Geological big data are generated regarding various layers of the earth, the history of the conformation and evolution of the earth, and the material composition of the earth and its changes. It also involves the exploration and utilization of mineral resources. In the current geological work, the collection, mining, processing, analysis, and utilization of various complex type data are closely related to those general big data. The "4V" characteristics of big data—namely, Volume, Velocity, Variety, and Veracity—also apply to geological big data.

3.1. Volume. Currently, there is no consensus on the size of geological data. Geological big data are a collection of data, including geology, minerals, remote sensing, geophysical exploration, geochemical exploration, surveying, and mapping, which are interconnected and integrated. In terms of the number of mines, there are at least 70000 in China, and some official documents and popular science books indicate that there are more than 200000 deposits and minerals that have been found. The information is huge and cannot be processed

using conventional tools. For example, an Excel spreadsheet cannot contain all the information of 70000 mining areas. Then, it is difficult to classify and rank the 200000 mines, so it is necessary to rely on the concepts and technologies of big data [17].

Especially in recent years, images, video, and other types of data have emerged on a large scale. With the application of 3D scanning and other devices, the data volume has been increasing exponentially. The ability to describe the data is more and more powerful, and the data are gradually approximated to the real world. In addition, the large amount of data is also reflected in the aspect that the methods and ideas used by people to deal with data have undergone a fundamental change. In the early days, people used the sampling method to process and analyze data in order to approximate the objective with a small number of subsample data. With the development of technologies, the number of samples gradually approaches the overall data. Using all the data can lead to a higher accuracy, which can explain things in more detail [18].

Recently, the China geological survey system has built databases including regional geological database (covering the 1 : 2500000, 1 : 1000000, 1 : 500000, 1 : 250000, and 1 : 200000 regional geological map; the national 1 : 200000 natural sand; the isotope geological dating; and the lithostratigraphic unit database), basic geological database (covering the national rock property database and national geological working degree database), mineral resources database (covering the national mineral resources, the national mineral resources utilization survey mining resources reserves verification results, the national survey of large and medium-sized mines, the prospect of mineral resources, the survey of the resources potential of major solid mineral resources in China, and the geological and mineral resources database), oil and gas energy database (covering the oil and gas basins in China, the geological survey results of the national oil and gas resources, the national petroleum and geophysical exploration, national shale gas, national coal bed methane, national natural gas hydrate, and other databases), geophysical database (covering 1 : 1 million, 1 : 500000, 1 : 250000, 1 : 200000, and 1 : 50000 gravity, national regional gravity, national aeromagnetism, national ground magnetism, national electrical survey, seismic survey, national aviation radioactivity, and national logging database), geochemical database (covering the databases of national 1 : 250000 and 1 : 20 geochemical exploration, national multiobjective geochemical and national land quality evaluation results), remote sensing survey database (covering national aeronautical remote sensing image, China resources satellite data, space remote sensing image, national mine environmental remote sensing monitoring, national high score satellite, and other databases), drilling database (covering the national geological borehole information, the national important geological borehole, the Chinese mainland scientific drilling core scanning image library, and so on), hydraulic cycle hazards database, data literature database, special subject database (covering the national mineral resources potential evaluation database, the important mineral "three-rate" investigation and evaluation database), work management database (covering the national exploration right, mining right, mining right verification, geological information metadata database, and many others) [17].

For those databases, they are still expanding and consummating, and their practical values have not yet been fully reflected. However, the vast majority of researchers are virtually impossible to have all of the above data, at most, using their own accumulated data. Anyway, even if their accumulated data, both on the quantity and on type, is incomparable by 10 years and 20 years ago, they are, in fact, in the era of the "relatively big data." From 1999 to 2004, for example, in "the Chinese mineralization system and regional metalorganic evaluation" project, although there are 202 national academic experts that participated in it, they only master data of 4500 properties (all kinds of minerals). From 2006 to 2013, the study of "national important mineral and regional mineralization laws" was conducted; meanwhile, the mining area covered only by the mineral resources research institute was 30600. Therefore, the increase of information and the amount of data are unprecedented in the last ten years.

3.2. Variety. From the formal point of view, the geological big data have many characteristics, including multidimensionality, multiscale, and multitenses. And they contain structured, semistructured, and unstructured data and usually are stored in forms of text, graphics, images, databases (including image database, spatial database, and attribute database), tables, videos, and audios in a fragmented state. For example, a large number of field outcrop description data, borehole core description data, and all kinds of geological survey, exploration report, and a large number of geological maps, drawings, and photos were stored and managed in the form of paper for a long time; even the numerous relational databases and spatial databases were primarily used to store and manage structured data that are tabulated and vectorized, while the text descriptions, records, and summaries were directly stored. Very few standardized processing and structural transformations were performed. Furthermore, there is no tool available to effectively integrate storage and manage structured, semistructured, and unstructured data.

3.3. Velocity. The increase of geological data is very fast, especially in remote sensing geology, aviation geophysical exploration, regional geochemical exploration, and other fields, due to the introduction of new technologies and new methods. Meanwhile, high speed processing is also a characteristic of big data. In addition to the need of analyzing data in real time, people also need to describe the results of data mining and processing through the use of several data processing techniques, such as image and video, while requiring effective and efficient handling skills. For example, the detection of the deep earth information not only needs to obtain parameters of the seismic wave reflection and refraction but also needs to conduct quick processing, so as to timely predict whether earthquake will occur and forecast the time, location, and intensity. In this way, we can avoid the disaster effectively. When applying a variety of data to a particular mountain, one should learn which ones have

spatial limitations and which are not related to spatiality, so that one deduces the metallogenic law and guides the prospecting better [17].

3.4. Veracity. For the understanding of the value of big data, most people consider it low value density. It means that the real useful information in the vast amount of data is very little. Taking video as an example, the useful data may be only a second or two in the continuous monitoring process. While big data is high value, it does not need to be invested too much; just collecting information from the Internet can bring business value. Therefore, big data has the characteristics of low value density and high business value. The same is true for geological big data. So far, there has been a lot of information about geophysical prospecting, but only a few have been confirmed, and the discovered mines were less. But once a breakthrough was made, its socioeconomic value was enormous, such as the lithium polymetallic deposit in Tibet and the newly discovered Jima copper polymetallic deposit in the outskirts of Sichuan [17].

In addition, the spatial attribute and temporal attribute of geological data also bring a big challenge to data accuracy. Any geological data have spatial attribute, and their values are reflected in the spatial law of distribution of mineral resources. For this reason, in the process of establishing the metallogenic series, exploring the metallogenic law, and constructing the mathematical model, the spatial attribute of the metallogenic model should be considered. Obviously, every metallogenic series has its spatial attribute. Geological data also has the time attribute, which is very different from physical, chemical, and other natural sciences. One of the fundamental pillars of geology is the geological time scale. The rocks, strata, and deposits of different geological periods have different distribution characteristics and regularity, so those data have their own time attribute.

It is obvious that those characteristics of geological big data mentioned above impose very challenging obstacles to the data management in CEGIS. The challenges related to geological big data management can be summarized as follows:

(i) It is quite difficult to describe and model geological big data, since there are few effective characteristics description mechanisms and object modeling approaches under the cloud computing environment.

(ii) There remain many technical issues that must be addressed to fully manage, mine, analyze, integrate, and share those geological big data, in consideration of those complex characteristics, including multisource heterogeneous data, highly spatiotemporal variation, high-volume and high-correlation data, and many others.

(iii) Many issues appear in achieving decision support, such as data incompleteness, data uncertainty, and high-dimensionality of data.

The broad range of challenges described here make good topics for research within the field of big data management in CEGIS. They are analyzed in the next section.

4. Key Technologies and Trends on Big Data Management in Cloud-Enabled Geological Information Service (CEGIS)

With the rapid advancement of big data technologies, some key technologies are accordingly developed for big data management in CEGIS. Specifically, a schematic diagram of those key technologies is shown in Figure 5. Then, in this section we present an analysis on those key technologies. Meanwhile, the trends along this direction are also discussed.

4.1. Geological Big Data Collection and Preprocessing. Geological big data collection and preprocessing aim to categorize those geological big data obtained from geological data, geological information, and geological literature.

4.1.1. Geological Data Collection Access. In addition to the traditional collection ways, it is also required to carry out large-scale network information access and provide real-time, high concurrency, and fast web content acquisition, combining with the application characteristics in the cloud environment. Currently, considering that the growth rate of geological information generated from the network is very fast, the big data analysis system should obtain relevant data quickly.

4.1.2. Quality and Usability Characteristics of Geological Data. It needs to distinguish and identify valuable information through intelligent discovery and management technologies. Because the information value density contained in different data sources differs from each other, filtering out the useless or low-value data source can effectively reduce the data storage and processing costs. Then, it can also further improve the efficiency and accuracy of analysis.

4.1.3. Geological Data Entity Recognition Model. According to the subject domain of geology, the distributed data are extracted to form a data warehouse, after conducting the operation of processing and integration. When extracting data in the field of geology, it needs to use entity modeling method to abstract entities from the vast numbers of data, so as to find out the relationship between those entities. This approach ensures that the data used in warehouse data can be consistent and relevant in accordance with the data model [19]. These recognized data are directly input into the system, stored as metadata, which could be used for data management and analysis.

4.1.4. Aggregation of Geological Big Data. Generally, different data sources and even the same data source may generate data with different formats. As mentioned above, because these structural, semistructured, and unstructured multimodal geological big data are integrated together, the data heterogeneity is obvious in big data analysis. Then, data aggregation as the key technology in achieving data extraction and transformation [20] enables data sharing and data fusion between heterogeneous data sources. Through

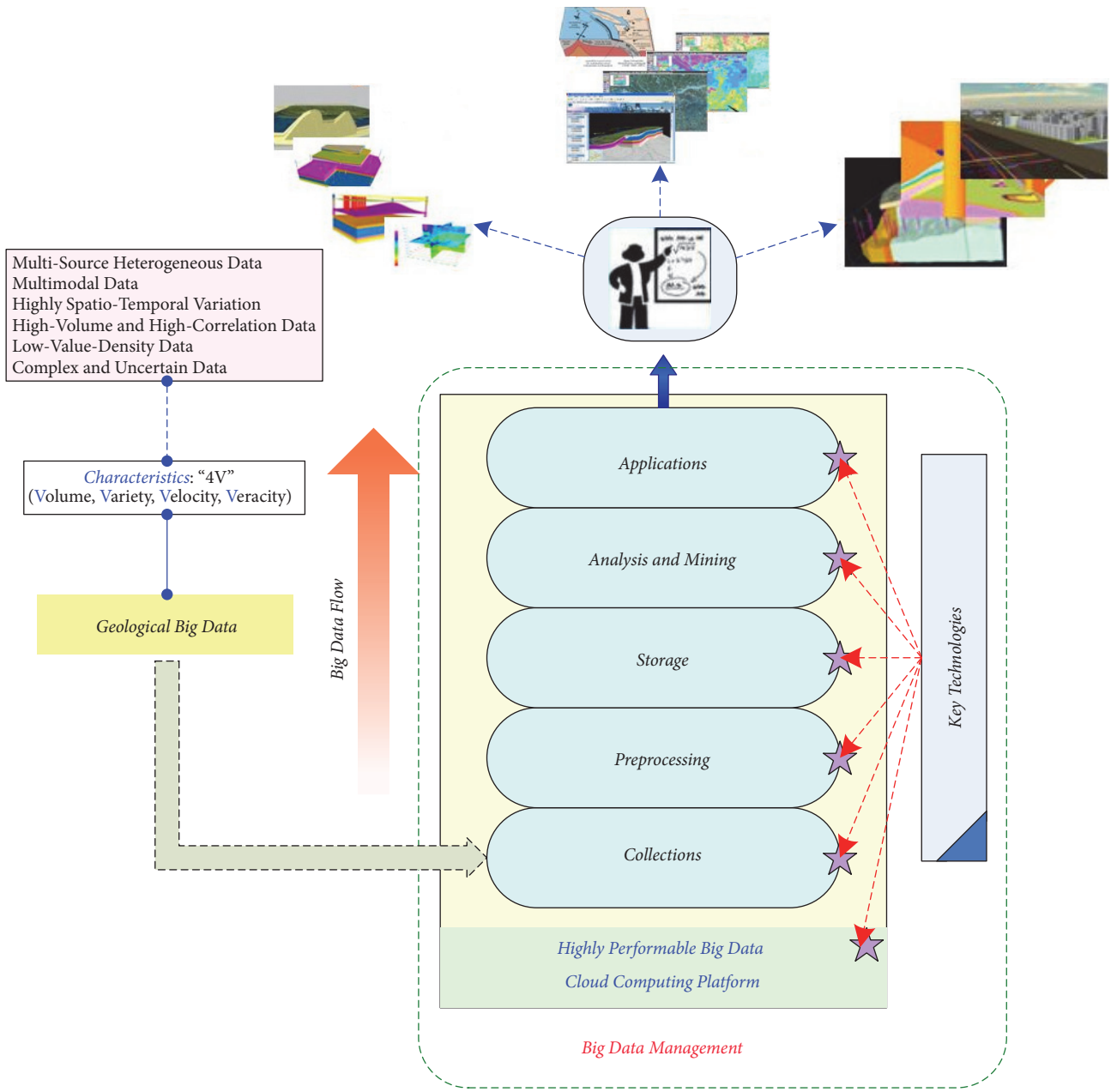


Figure 5: Schematic diagram of key technologies for big data management in CEGIS.

the use of heterogeneous information aggregation technologies, the unified data retrieval and data presentation could be achieved. On the basis of it, after aggregating those distributed heterogeneous data sources, they are extracted and converted to achieve the functions of automatically constructing subject domain database and data warehouse [21].

4.1.5. Management of Geological Big Data Evolution Tracking Records. In order to effectively utilize geological big data, it needs to track the evolution of big data during the whole life cycle of GIS, with the purpose of achieving the traceable big data management.

Here, we provide an example of aggregating and collecting geological big data in CEGIS. Figure 6 illustrates this process. While developing CEGIS, all kinds of geological data should be processed. Through the use of geological cloud, big data are collected, and then they are aggregated to achieve some key functions in geological information service platform, including catalog sharing, intelligent searching, data products release, and collaborative service.

4.2. Geological Big Data Storage and Management. From the data collection perspective, geological data can be divided into field survey data, drilling and engineering exploration data, remote detection data, analytical test data, and

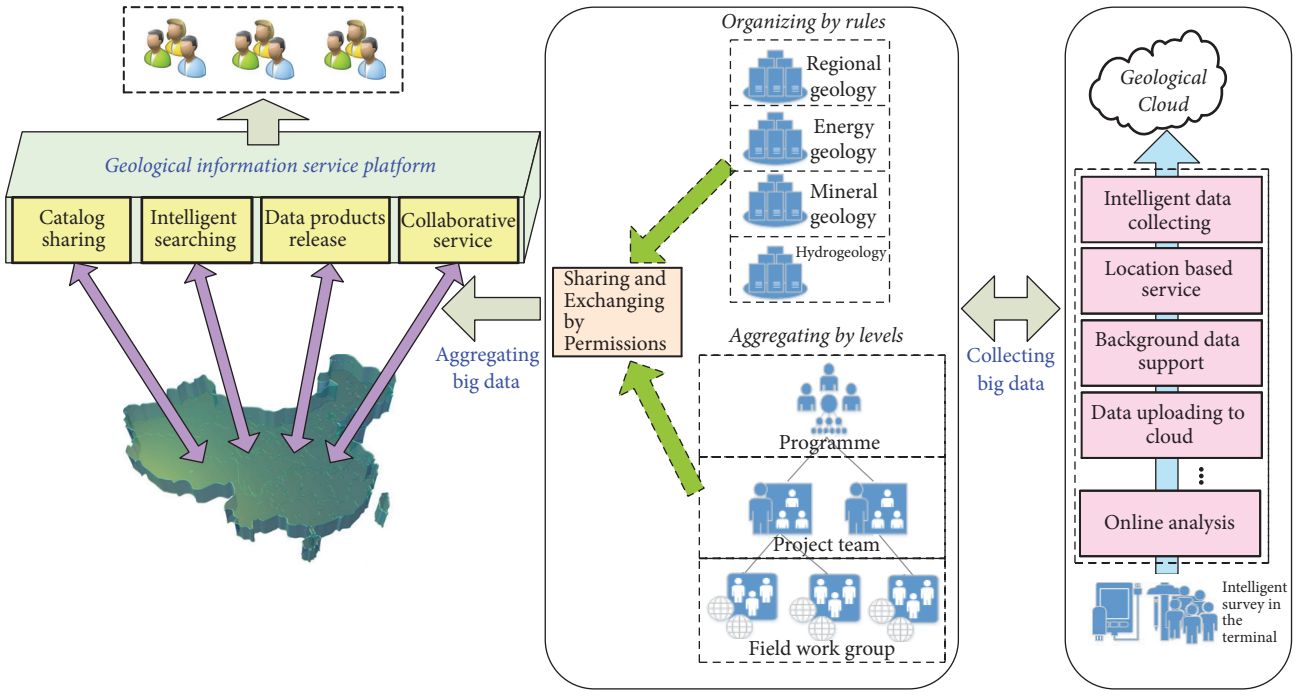


Figure 6: An example of aggregating and collecting geological big data in CEGIS.

comprehensive study data. From the angle of comprehensive application fields, they can be also divided into regionally geological survey data, energy and mineral resources evaluation and exploration survey results data, geological disaster monitoring and early warning data, geological environment survey and evaluation results data, and marine geological survey and evaluation data. From the data formality point of view, they can be divided into picture data, text report data, tabular data, and image data. These data are collected by different units.

Facing these complex geological big data mentioned above, the traditional relational database will be difficult to handle them, while the distributed storage system can be used to store such huge amounts of data and manage them. Then, the data system places the massive data in many machines, which avoids such limitation of storage capacity, and also brings many problems that have not occurred before in standalone systems. Hence, some distributed data storage solutions have accordingly emerged, including Hadoop, Spark, and other nonrelational database systems (like HBase, MongoDB, and many others) [22]. These different solutions satisfy the specific requirements from different applications. When applying to the analysis of big data, different solutions can be employed according to the specific needs of different intelligence analysis. Furthermore, different solutions can be combined to meet specific needs. Actually, there have been some attempts to develop combination strategies for distributed storage model, varying in the big data management performance requirement, and the complexity of collected big data that are supported by the distributed storage system [23]. Hence, there is still a room for improvement and optimization of geological big data storage, while designing a hybrid distributed storage model through the use of cloud advantages of flexibly scalable deployment, to meet the users' requirement for geological big data resource management with satisfactory data durability and high availability [23].

Here, the hot research topics include the following:

(i) For geological applications, the load optimization storage should be implemented to achieve the coupling for data storage and application and the coupling for distributed file system and the new storage system.

(ii) Based on the application characteristics of distributed databases, more studies could be conducted on the application of new databases NoSQL and NewSQL in geological survey work.

With the development of big data technologies, more and more mature distributed data storage solutions will emerge and will be applied to big data analysis [24, 25].

Specifically, in the management of geological big data, the implementation of data query—for example, spatial query—has been a long-term focus. Generally, considering those advantages with unified modeling language (UML) and computer-aided software engineering (CASE) methodology, the spatial database could be accordingly designed and implemented to characterize and realize the object-oriented spatial vector big data firstly [26]. And then, in the developed spatial database, the function of self-generating codes would be achieved to realize two-way spatial query between graphic-objects and property data [26]. Moreover, in consideration of the complex characteristics of geological big data, the spatial query is achieved finally through the use of Flex technology in ArcGIS Server software platform [27]. Practically speaking,

in this technology, the spatial query could be implemented through two functions, including "Query" and "Find" query methods [28].

4.3. Geological Big Data Analysis and Mining. In terms of geological data analysis and mining, it needs to combine geological data, geological information, and geological literatures, through the analysis of geological application demand of real-time mining, to explore geological big data environment analysis and mining algorithm, in an effort to fully achieve the goal of intelligent mining for geological big data.

Figure 7 shows a schematic diagram of discovering geological knowledge through analyzing and mining geological big data. It can be easily found that geological big data analysis and mining play an important role in achieving the final goal. More relevant research work related to it mainly involves the following aspects.

4.3.1. Geological Big Data Analysis. Considering the special applications, geological big data technologies would apply big data concepts to analyze the metallogenic rules by making full use of various data related to ore, to recognize deposit metallogenic series, to summarize the metallogenic regularities and express in an appropriate way (like voice, image, and many others), and to establish the scientifically mathematical model. The model then uses new exploration data to predict future data and to guide geological prospecting.

In addition, it is necessary to pay special attention to the analysis of new geological big data information collected from social medium and networks [29]. These include the geological text information flow data from microblog web sites, the geological multimedia data from media sharing web sites, the geology-related user interaction data on social networking web sites, and many others [30]. These multisource data complement traditional big data. Specifically, such data should be addressed with the help of multilingual information processing, multilingual machine translation, and social network cross-language retrieval [31]. Big data analysis of such data is a key to deep use of geological data in a broader dimension. With the maturity of big data analysis technologies, it becomes possible to analyze and extract valuable information from these data [32] and to provide effective solutions for geological big data applications.

4.3.2. Geological Big Data Mining. Data mining is to extract the unknown and useful knowledge and information from the massive multilevel spatiotemporal data and attribute data, using statistics, pattern recognition, artificial intelligence, set theory, fuzzy mathematics, cloud computing, machine learning, visualization, and relevant techniques and methods. Data mining could reveal the relationship and evolution trend behind the geological big data, achieve the automatic or semiautomatic acquisition of the new knowledge, and provide the decision basis for resource prediction, prospecting, environmental assessment, and disaster prevention and mitigation [33]. Therefore, the knowledge is obtained directly from known geological data to provide relevant decision support [34]. In consideration of the amount of data, it may deal with terabytes or even petabytes of data, as well as multidimensional data, all kinds of noise data and dynamic data. Because data mining algorithms will directly influence the outcome of the discovered knowledge, selecting the most appropriate algorithms and parallel computing strategy is the key to data mining.

Effective data mining also could reduce manual intervention during information processing and make use of methods and tools of big data intelligent analysis [35, 36]. Recently, there has been a growing interest in the geological big data mining through the use of some novel computational intelligent methods—for example, rough set [37] and fuzzy aggregation [38]. Moreover, with the development of those neural network based machine learning algorithms in recent years, some popular methods, including extreme learning machine [39, 40], approximate dynamic programming [41], and kernel learning [42], could be used to further improve mining effectiveness for geological big data in the future.

4.4. Highly Performable Big Data Cloud Computing Platform. Highly performable big data cloud computing platform is the foundation for big data analysis. It enables parallel computing for large-scale incremental real-time data and large-scale heterogeneous data [43–46].

With the advent of massive data storage solution, many big data distributed computing frameworks have been proposed. Among them, Hadoop, MapReduce, Spark, and Storm are the most important distributed computing frameworks. These frameworks have different characteristics and solve different problems in applications [47–50]. The Hadoop/MapReduce is often used for offline complex big data processing, the Spark is often employed in offline fast big data processing, and the Storm is often available for real-time online big data processing. Different computing frameworks have their different advantages and disadvantages. Hadoop/MapReduce is easy to program, and it is with satisfactory scalability and fault tolerance. In addition, it is suitable for offline processing of massive data with petabyte level, but it does not support real-time computation and flow calculation. Spark is a memory-based iterative computing framework. By placing intermediate data in memory, Spark can achieve higher iterative calculation performance. The programming model of Spark is more flexible than that of Hadoop/MapReduce, but Spark is not suitable for those applications in which the fine-grained updates are conducted asynchronously. Hence, Spark may be unavailable for those application models that require incremental changes. Storm is suitable for stream data processing. It can be used to handle a stream of incoming messages and can write the processed result to a specified storage device. Another major application of Storm is real-time data processing where data are not necessary to be written into storage devices, which usually results in low time delay. Hence, Storm is particularly suitable for scenarios where real-time online analysis is required to obtain results for big data analysis.

An application example is geological big data aggregation mining framework based on Hadoop [16]. Geological big data aggregation mining platform research is based on the

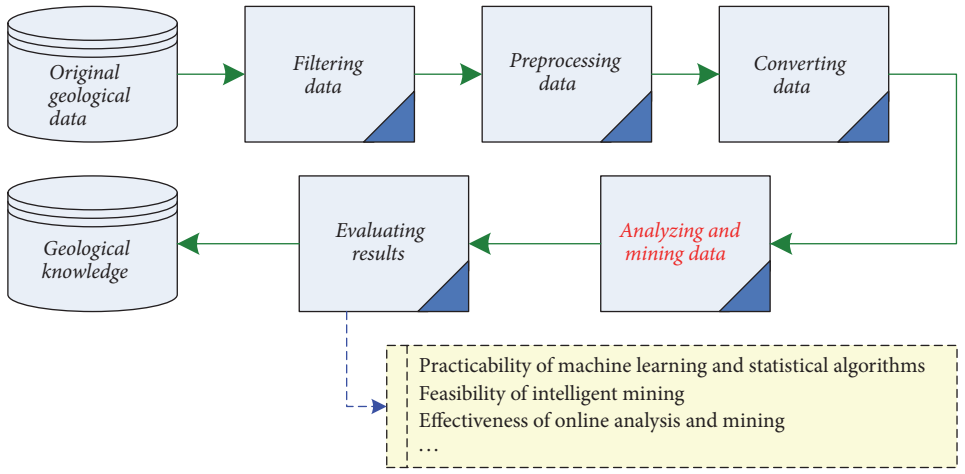


Figure 7: Schematic diagram of discovering geological knowledge through analyzing and mining geological big data.

China geological survey data network, and it uses the Hadoop technology to improve and modify existing platform, to make it suitable for big data applications, and to provide a platform for the pilot applications. The geological survey grid platform can be updated in three layers—that is, the virtual layer, the computing layer, and the terminal application layer. The virtual layer is the virtualization of computer resources based on Hadoop distributed file system (HDFS) virtualization technology, which is the foundation of cloud computing and cloud services. The computing layer mainly uses MapReduce method to implement the analysis algorithms for geological big data. Currently, the geological big data technologies mainly use the block calculation strategy to achieve parallel analysis through the utilization of the characteristics of Hadoop, in an effort to speed up the analysis and processing of geological data. The terminal application layer is designed to display the results and receive user feedback to improve system availability.

MapReduce has been used to perform morphological correlation analysis, which involves the analysis of geochemical data processing and the study of the correlation between multielements. Figure 8 shows the pattern correlation between elements. It can be seen from Figure 8 that the elements of Mn, Co, and Be are similar in the distribution of morphology. Therefore, from a qualitative point of view, the correlation is relatively high. Moreover, after testing, the proposed prototype system is running three times more quickly than the existing common computing platform, showing that the geological big data is applicable to the Hadoop platform. Furthermore, some applications of using MapReduce could be found in [51].

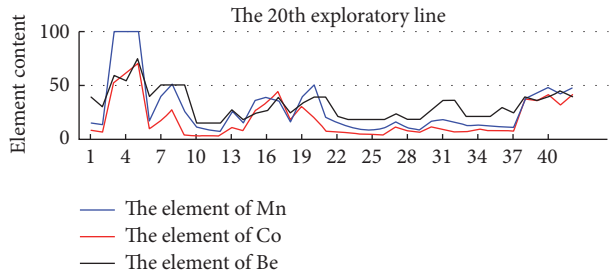


Figure 8: Correlation among three element morphologies.

4.5. Applications of Geological Big Data Technologies

4.5.1. Exploration of Metallogenic Law. The metallogenic law is the human regular knowledge of the temporal and spatial distribution of mineral resources, and its cognitive level, ability, and scope are all related to the size of data, the type of data, and the way of data processing. Therefore, to deduce the metallogenic law, it is necessary to fully understand the massive data about spatial distribution, reserves and production in mineral origin, the geological structure of the mineral origin, and related geological survey data. Then, it is to conduct the regular speculation and objectivity expression of these geological big data, so that one can identify the essential reasons for the distribution of mineral origin. Actually, using geological big data technologies could help to translate data into new understanding or knowledge and help to guide the future geological prospecting work.

4.5.2. Smart Prospecting. The types of deposits vary, and the formation of them is related to certain geological backgrounds and geological effects, respectively. The geological backgrounds include tectonic unit and stratigraphic unit, deep upper mantle and lithosphere conditions, and paleogeography and palaeoclimate environment on the surface of the earth. Geological effects include tectonism, magmatism, sedimentation, metamorphism, and weathering. Theses geologic backgrounds and effects in the wide range of space, and in the long geologic history, are a dynamic change and repeated stack, and large deposits can be formed only in a variety of favourable conditions. Long-term scientific research and experience accumulation formed mineral deposit and mineralization prediction subject. Professionals

are guided by certain theories and methods to adopt quantitative or qualitative methods to predict prospecting with the existing knowledge and experience.

However, in view of the difficulties of geological data sharing and the limitations of calculation tools and calculation methods, most of the known deposits in the past are independent of each other. In the future, we can use geological data to connect several adjacent deposit exploration data, conduct unified analysis and specialized processing, determine the "digital" characteristics of the distribution of metallogenic materials, find out metallogenic potential, delineate the abnormal area and prospective area, and promote geological prospecting. Furthermore, geological data informatization and standardization could be improved [52].

4.5.3. Service of People's Livelihood Geology. After entering the 21st century, geological work is more closely related to economic development, and geological work plays an important role in every aspect of social and economic life. Agricultural geology, urban geology, environmental geology, tourism geology, disaster geology, and other works have been strengthened, and the service area has also been expanded [53]. Meanwhile, the public demand for geological information is increasingly urgent [54].

In order to meet the social demand for geological data, China Geological Survey carried out the construction of geological cloud, which built cluster geologic data service system with the National Geological Information Center and the Provincial Geological Information Center as the backbone nodes, conducted the integration of data resources, and applied the GIS cloud technology, in order to obtain large-scale computing ability and solve those key problems, such as the distributed storage, processing, query, interoperability, and virtualization of massive spatial data [5, 13]. Recently, in China, Shandong Provincial Bureau of Geology and Mineral Resources also carried out the construction of "the application system of geological business based on e-government cloud platform." It mainly relies on the public service cloud platform of the e-government in Shandong province and constructs the government external network service system and Internet service system to achieve the unified management and information service of the mineral resource. Using technical methods of spatial analysis, big data mining, and three-dimensional geological model, it develops a basic system framework for geological mining services, featured by "a (cloud) platform, a (data) center, and many application systems," to improve the ability of the people's livelihood geological service, promote interaction with the public, realize socialization services, and promote the clustering and industrialization of the mineral resources information services.

4.5.4. Application of Knowledge Visualization Service. With the continuous development of web technology, human beings have experienced the "Web 1.0" era, which is characterized by document interconnection, and "Web 2.0", which is characterized by data interconnection, and are moving towards the new "Web 3.0" era based on the interconnected knowledge of the entity. Due to the continuous release of user-generated content and linking open data on the Internet, people need to explore knowledge interconnection methods which both conform to the development of the network information resources and meet users' requirements from a new perspective according to the knowledge organization principles in the large data environment, to reveal human cognition on a deeper level [55].

In this context, knowledge graph (KG) was formally put forward by Google in May 2012, and its goal is to improve the search results and describe the various entities and concepts that exist in the real world and the relationship between these entities and concepts. KG is a great choice to select the essence and discard the dross, as well as the sublimation of the present semantic web technology. In recent years, the applications of KG have been increasing rapidly, and there is now a mature method used to draw a KG and conduct intelligent searching research based on KG [34]. However, the function of KG has not been fully implemented at present, especially for the specific object of geological big data; the application aspect still needs to be further strengthened. Along this direction, the visualization service for geological data in the web-based system is attracting more and more attention [56, 57].

5. Conclusion

Big data technologies make it possible to process massive amount of unstructured and semistructured geological data. And the geological cloud enables us to explore the application of demand-driven geological core data and to extract new information from unstructured data, while supporting the decision-making in land resources management. Thus, the geological cloud could effectively organize and use geological big data, to mine the data scientifically, with the purpose of producing higher value and achieving the corresponding service.

In the architecture of geological cloud, this article describes the application background of CEGIS and the demands from big data management. Furthermore, we elaborate the application requirements and challenges faced in big data management technologies. Then, more analyses are provided from four aspects, including data size, data type, data processing speed, and data processing accuracy, respectively. In addition, this article outlines the research status and technology development opportunities of big data related in CEGIS, from the perspectives of big data acquisition and preprocessing, big data storage and management, big data analysis and mining, highly performable big data cloud computing platform, and big data technology applications. With the continuous development of big data technologies in addressing those challenges related to geological big data, such as the difficulties of describing and modeling geological big data with some complex characteristics, CEGIS will move towards a more mature and more intelligent direction in the future.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work was supported in part by the Key Laboratory of Geological Information Technology of Ministry of Land and Resources under Grant 2017320, the National Key Technologies R&D Program of China under Grant 2015BAK38B01, and the National Key R&D Program of China under Grant 2016YFC0600510.

References

[1] P. Vermeesch and E. Garzanti, "Making geological sense of 'big data' in sedimentary provenance analysis," *Chemical Geology*, vol. 409, pp. 20–27, 2015.

[2] J. Chen, J. Xiang, Q. Hu et al., "Quantitative geoscience and geological big data development: a review," *Acta Geologica Sinica*, vol. 90, no. 4, pp. 1490–1515, 2016.

[3] Y. Zhu, Y. Tan, R. Li, and X. Luo, "Cyber-Physical-Social-Thinking modeling and computing for geological information service system," *International Journal of Distributed Sensor Networks*, vol. 12, no. 11, 2016.

[4] M. M. Song, Z. Li, B. Zhou, and C. L. Li, "Cloud computing model for big geological data processing," *Applied Mechanics and Materials*, vol. 475-476, pp. 306–311, 2014.

[5] J. P. Chen, J. Li, N. Cui, and P. P. Yu, "The construction and application of geological cloud under the big data background," *Geological Bulletin of China*, vol. 34, no. 7, pp. 1260–1265, 2015.

[6] C. Li, "The technical infrastructure of geological survey information grid," in *Proceedings of the 18th International Conference on Geoinformatics*, pp. 1–6, 2010.

[7] L. Wu, L. Xue, C. Li et al., "A geospatial information grid framework for geological survey," *PLoS ONE*, vol. 10, no. 12, Article ID e0145312, 2015.

[8] K. Evangelidis, K. Ntouros, S. Makridis, and C. Papatheodorou, "Geospatial services in the Cloud," *Computers and Geosciences*, vol. 63, no. 2, pp. 116–122, 2014.

[9] M. Huang, A. Liu, T. Wang, and C. Huang, "Green data gathering under delay differentiated services constraint for internet of things," *Wireless Communications and Mobile Computing*, 2018, http://downloads.hindawi.com/journals/wcmc/aip/9715428.pdf.

[10] https://www.webofknowledge.com/.

[11] C. Yang, M. Yu, F. Hu, Y. Jiang, and Y. Li, "Utilizing Cloud Computing to address big geospatial data challenges," *Computers, Environment and Urban Systems*, vol. 61, pp. 120–128, 2017.

[12] L. Wu, L. Xue, C. Li et al., "A knowledge-driven geospatially enabled framework for geological big data," *ISPRS International Journal of Geo-Information*, vol. 6, no. 6, article no. 166, 2017.

[13] Y. Tan, "Architecture and key issues of geological big data and information service project," *Geomatics World*, vol. 23, no. 1, pp. 1–9, 2016.

[14] Y. Tan, "Architecture investigation of the construction of geological big data system," *Geological Survey of China*, vol. 3, no. 3, pp. 1–6, 2016.

[15] W. He and Y. Wang, "Prototype system of geological cloud computing," *Progress in Geophysics*, vol. 29, no. 6, pp. 2886–2896, 2014.

[16] Y. Zhu, Y. Tan, J. Zhang, B. Mao, J. Shen, and C. Ji, "A framework of hadoop based geology big data fusion and mining technologies," *Acta Geodaetica et Cartographica Sinica*, vol. 44, no. S0, pp. 152–159, 2015.

[17] D. Wang, X. Liu, and L. Liu, "Characteristics of big geodata and its application to study of minerogenetic regularity and minerogenetic series," *Mineral Deposits*, vol. 34, no. 6, pp. 1143–1154, 2015.

[18] B. Pan and R. Yang, "Management and utilization of big data for geology," *Surveying and Mapping of Geology and Mineral Resources*, vol. 33, no. 1, pp. 1–3, 2017.

[19] P. Yang and L. J. Lu, "The research on encoding methodology of the character of geological entity based on mass geological data," *Advanced Materials Research*, vol. 962-965, pp. 208–212, 2014.

[20] X. Luo, D. Zhang, L. T. Yang, J. Liu, X. Chang, and H. Ning, "A kernel machine-based secure data sensing and fusion scheme in wireless sensor networks for the cyber-physical systems," *Future Generation Computer Systems*, vol. 61, pp. 85–96, 2016.

[21] C.-L. Kuo and J.-H. Hong, "Interoperable cross-domain semantic and geospatial framework for automatic change detection," *Computers & Geosciences*, vol. 86, pp. 109–119, 2016.

[22] Y.-J. Wang, W.-D. Sun, S. Zhou, X.-Q. Pei, and X.-Y. Li, "Key technologies of distributed storage for cloud computing," *Journal of Software* , vol. 23, no. 4, pp. 962–986, 2012.

[23] M. Armbrust, A. Fox, R. Griffith et al., "Above the clouds: a berkeley view of cloud computing," Tech. Rep. UCB/EECS-2009-28, University of California at Berkeley, California, Calif, USA, 2009.

[24] J. Xia, Z. Bai, B. Wang, J. Chang, and Y. Wu, "Design and implementation of comprehensive management platform for geological data informatization," *Acta Scientiarum Naturalium Universitatis Pekinensis*, vol. 50, no. 2, pp. 295–300, 2014.

[25] W. Hua, J. Liu, and X. Liu, "Data management of object type geological features on control dictionary," *Earth Science - Journal of China University of Geosciences*, vol. 40, no. 3, pp. 425–430, 2015.

[26] B. Jia, C. Wang, C. Liu, and W. W. Sun, "Design and implementation of object-oriented spatial database of coalfield geological hazards -Based on object-oriented data model," in *Proceedings of the 2010 International Conference on Computer Application and System Modeling, ICCASM '10*, pp. V1282–V1286, Taiyuan, China, IEEE, October 2010.

[27] http://server.arcgis.com/.

[28] X. Zhou, X. Li, A. Chen et al., "Design and implementation of the service system of spatial data for geological data," *Journal of Geomatics*, vol. 38, no. 4, pp. 57–60, 2013 (Chinese).

[29] H. Huang, Z. Chao, and C. Feng, "Opportunities and challenges of big data intelligence analysis," *CAAI Transactions on Intelligent Systems*, vol. 11, no. 6, pp. 719–727, 2016.

[30] S. Jin, W. Lin, H. Yin, S. Yang, A. Li, and B. Deng, "Community structure mining in big data social media networks with MapReduce," *Cluster Computing*, vol. 18, no. 3, pp. 999–1010, 2015.

[31] C. C. Yang, C.-P. Wei, and L.-F. Chien, "Managing and mining multilingual documents: Introduction to the special topic issue of information processing management," *Information Processing & Management*, vol. 47, no. 5, pp. 633-634, 2011.

[32] X. Luo, J. Deng, W. Wang, J.-H. Wang, and W. Zhao, "A quantized kernel learning algorithm using a minimum kernel risk-sensitive loss criterion and bilateral gradient technique," *Entropy*, vol. 19, no. 7, article 365, 2017.

[33] C. H. Tse, Y. L. Li, and E. Y. Lam, "Geological applications of machine learning in hyperspectral remote sensing data," in *Proceedings of Conference on Image Processing - Machine Vision Applications VIII*, 2015.

[34] Y. Zhu, W. Zhou, Y. Xu, J. Liu, and Y. Tan, "Intelligent learning for knowledge graph towards geological data," *Scientific Programming*, vol. 2017, Article ID 5072427, 13 pages, 2017.

[35] H. X. Vo and L. J. Durlofsky, "Data assimilation and uncertainty assessment for complex geological models using a new PCA-based parameterization," *Computational Geosciences*, vol. 19, no. 4, pp. 747–767, 2015.

[36] A. Gasmi, C. Gomez, H. Zouari, A. Masse, and D. Ducrot, "PCA and SVM as geo-computational methods for geological mapping in the southern of Tunisia, using ASTER remote sensing data set," *Arabian Journal of Geosciences*, vol. 9, no. 20, article 753, 2016.

[37] Z.-S. Luo and Y.-T. Wei, "Research on Rough set applied in the geological measure data prediction model," *Advanced Materials Research*, vol. 457-458, pp. 792–798, 2012.

[38] M. Farzamian, A. K. Rouhani, A. Yarmohammadi, H. Shahi, H. A. F. Sabokbar, and M. Ziaiie, "A weighted fuzzy aggregation GIS model in the integration of geophysical data with geochemical and geological data for Pb–Zn exploration in Takab area, NW Iran," *Arabian Journal of Geosciences*, vol. 9, no. 2, article no. 104, pp. 1–17, 2016.

[39] Y. Xu, X. Luo, W. Wang, and W. Zhao, "Efficient DV-HOP localization forwireless cyber-physical social sensing system: a correntropy-based neural network learning scheme," *Sensors*, vol. 17, no. 1, article 135, 2017.

[40] X. Luo, Y. Xu, W. Wang et al., "Towards enhancing stacked extreme learning machine with sparse autoencoder by correntropy," *Journal of The Franklin Institute*, 2017.

[41] X. Luo, H. Luo, and X. Chang, "Online optimization of collaborative web service qos prediction based on approximate dynamic programming," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 452492, 9 pages, 2015.

[42] X. Luo, J. Deng, J. Liu, W. Wang, X. Ban, and J. Wang, "A quantized kernel least mean square scheme with entropy-guided learning for intelligent data analysis," *China Communications*, vol. 14, no. 7, pp. 127–136, 2017.

[43] J. Passmore, J. Laxton, and M. Sen, "EarthServer for geological applications opening up access to big data using OGC web services," *Advances in Soil Mechanics and Geotechnical Engineering*, vol. 3, pp. 123–129, 2014.

[44] C. Li, M. Song, L. Xia, X. Luo, and J. Li, "The spatial data sharing mechanisms of geological survey information grid in P2P mixed network systems network architecture model," in *Proceedings of the 9th International Conference on Grid and Cloud Computing, GCC '10*, pp. 258–263, November 2010.

[45] S. A. B. Cruz, A. M. V. Monteiro, and R. Santos, "Automated geospatial Web Services composition based on geodata quality requirements," *Computers & Geosciences*, vol. 47, pp. 60–74, 2012.

[46] J. Xia, C. Yang, K. Liu, Z. Li, M. Sun, and M. Yu, "Forming a global monitoring mechanism and a spatiotemporal performance model for geospatial services," *International Journal of Geographical Information Science*, vol. 29, no. 3, pp. 375–396, 2015.

[47] S. Ibrahim, H. Jin, L. Lu, L. Qi, S. Wu, and X. Shi, "Evaluating MapReduce on virtual machines: the hadoop case," in *Proceedings of the First International Conference on Cloud Computing*, pp. 519–528, 2009.

[48] M. H. Iqbal and T. R. Soomro, "Big data analysis: apache Storm perspective," *International Journal of Computer Trends and Technology*, vol. 19, no. 1, pp. 9–14, 2015.

[49] J. L. Reyes-Ortiz, L. Oneto, and D. Anguita, "Big data analytics in the cloud: spark on Hadoop vs MPI/OpenMP on Beowulf," *Procedia Computer Science*, vol. 53, no. 1, pp. 121–130, 2015.

[50] X. Meng, J. Bradley, B. Yavuz et al., "MLlib: machine learning in apache spark," *Journal of Machine Learning Research*, vol. 17, 2016.

[51] R. Giachetta, "A framework for processing large scale geospatial and remote sensing data in MapReduce environment," *Computers and Graphics*, vol. 49, pp. 37–46, 2015.

[52] S. Huang and X. Liu, "Geological data informatization and standardization based on geological big data," *Coal Geology of China*, vol. 28, no. 7, pp. 74–78, 2016.

[53] K. J. A. Kouame, F. Jiang, Y. Feng, and S. Zhu, "The strengthening of geological infrastructure, research and data acquisition - using gis in ivory coast gold mines," *MATEC Web of Conferences*, vol. 95, p. 18001, 2017.

[54] C. S. J. Karlsson, S. Miliutenko, A. Björklund, U. Mörtberg, B. Olofsson, and S. Toller, "Life cycle assessment in road infrastructure planning using spatial geological data," *The International Journal of Life Cycle Assessment*, vol. 22, no. 8, pp. 1302–1317, 2017.

[55] K. Stock, T. Stojanovic, F. Reitsma et al., "To ontologise or not to ontologise: an information model for a geospatial knowledge infrastructure," *Computers & Geosciences*, vol. 45, pp. 98–108, 2012.

[56] J. Hunter, C. Brooking, L. Reading, and S. Vink, "A Web-based system enabling the integration, analysis, and 3D sub-surface visualization of groundwater monitoring data and geological models," *International Journal of Digital Earth*, vol. 9, no. 2, pp. 197–214, 2016.

[57] R. D. Müller, X. Qin, D. T. Sandwell et al., "The GPlates portal: Cloud-based interactive 3D visualization of global geophysical and geological data in a web browser," *PLoS ONE*, vol. 11, no. 3, Article ID e0150883, 2016.

Hindawi
Scientific Programming
Volume 2018, Article ID 6749561, 8 pages
https://doi.org/10.1155/2018/6749561

Research Article

Incremental Graph Pattern Matching Algorithm for Big Graph Data

Lixia Zhang[1] and Jianliang Gao[2]

[1]*College of Mathematics and Computer Science, Key Laboratory of High Performance Computing and Stochastic Information Processing, Ministry of Education of China, Hunan Normal University, Changsha 410081, China*
[2]*School of Information Science and Engineering, Central South University, Changsha 410083, China*

Correspondence should be addressed to Jianliang Gao; gaojianliang@csu.edu.cn

Received 19 October 2017; Accepted 20 December 2017; Published 22 January 2018

Academic Editor: Longxiang Gao

Graph pattern matching is widely used in big data applications. However, real-world graphs are usually huge and dynamic. A small change in the data graph or pattern graph could cause serious computing cost. Incremental graph matching algorithms can avoid recomputing on the whole graph and reduce the computing cost when the data graph or the pattern graph is updated. The existing incremental algorithm PGC_IncGPM can effectively reduce matching time when no more than half edges of the pattern graph are updated. However, as the number of changed edges increases, the improvement of PGC_IncGPM gradually decreases. To solve this problem, an improved algorithm iDeltaP_IncGPM is developed in this paper. For multiple insertions (resp., deletions) on pattern graphs, iDeltaP_IncGPM determines the nodes' matching state detection sequence and processes them together. Experimental results show that iDeltaP_IncGPM has higher efficiency and wider application range than PGC_IncGPM.

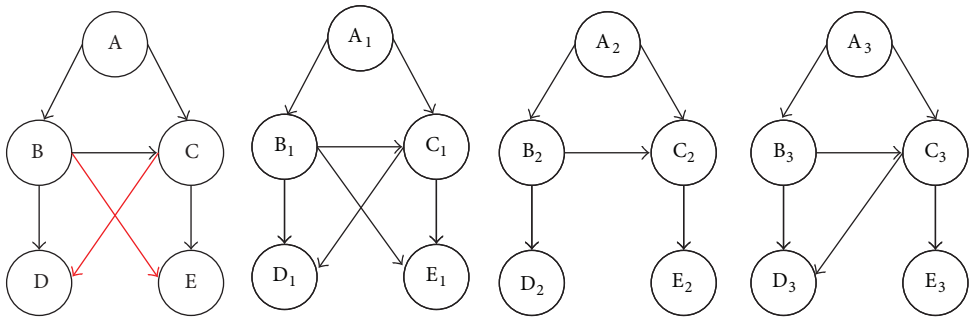
1. Introduction

Graph pattern matching is to find all the subgraphs that are the same or similar to a given pattern graph P in a data graph G. It is widely used in a number of applications, for example, web document classification, software plagiarism detection, and protein structure detection [1–3].

With the rapid development of Internet, huge amounts of graph data emerge every day. For example, the Linked Open Data Project, which aims to connect data across the Web, has published 149 billion triples until 2017 [4]. In addition, real-world graphs are dynamic [5]. It is often cost-prohibitive to recompute matches starting from scratch when G or P is updated. An incremental matching algorithm is needed, which aims to minimize unnecessary recomputation by analyzing and computing the changes of matching result in response to updates ΔG (resp., ΔP) to G (resp., P).

For example, Figure 1(a) is a pattern graph P and Figure 1(b) is a data graph G. The subgraph which is composed of A_1, B_1, C_1, D_1, E_1, and the edges between them (for simplicity, denoted as $\{A_1, B_1, C_1, D_1, E_1\}$) is the only matching subgraph. Assuming that (B, E) and (C, D) are removed from the pattern graph, the traditional recomputing algorithm will compute the matches for the new pattern graph on the whole data graph. It is time consuming. The incremental algorithm will just check a part of nodes in G, that is, B_2, B_3, C_2, C_3, A_2, and A_3, and add new matching subgraphs ($\{A_2, B_2, C_2, D_2, E_2\}$, $\{A_3, B_3, C_3, D_3, E_3\}$) to the original matching result.

At present, the study of incremental graph pattern matching is still in its infancy and existing work [6–12] mainly focuses on the updates of data graphs. In our previous study, we proposed an incremental graph matching algorithm named PGC_IncGPM, which can be used in scenarios where data graphs are constant and pattern graphs are updated [13]. PGC_IncGPM can effectively reduce the runtime of graph matching as long as the number of changed edges is less than the number of unchanged edges in P. However, the improvement effect of PGC_IncGPM gradually decreases as the number of changed edges increases. In this paper, the bottleneck of PGC_IncGPM is further analyzed. An optimization method of nodes' matching state detection sequence is proposed, and a more efficient algorithm called iDeltaP_IncGPM is designed and implemented.



(a) Pattern graph (b) Data graph

Figure 1: An example of incremental graph pattern matching.

Using Figure 1 as an example, suppose (B, E) and (C, D) are deleted from the pattern graph. PGC_IncGPM algorithm will first consider the deletion of (B, E), that is, checking B_2, A_2, B_3, and A_3, and then consider the deletion of (C, D), that is, checking C_2, B_2, A_2, C_3, B_3, and A_3. Thus B_2, A_2, B_3, and A_3 are all checked twice. iDeltaP_IncGPM considers the two deletions together; C_2, B_2, A_2, C_3, B_3, and A_3 are all checked only once.

The remainder of this paper is organized as follows. In Section 2, related work is reviewed. The model and definition are described in Section 3. In Section 4, our algorithm is presented. Section 5 is experimental results and comparison, and Section 6 presents the conclusion.

2. Related Work

We surveyed related work in two categories: graph pattern matching models and incremental algorithms for graph matching on massive graphs.

Graph pattern matching is typically defined in terms of subgraph isomorphism [14, 15]. However, subgraph isomorphism is an NP-complete problem [16]. In addition, subgraph isomorphism is often too restrictive because it requires that the matching subgraphs have exactly the same topology as the pattern graph. These hinder its applicability in emerging applications such as social networks and crime detection. Thus, graph simulation [17] and its extensions [18–22] are adopted for pattern matching. Graph simulation preserves the labels and the child relationship of a graph pattern in its match. In practical applications, graph simulation is so loosely that it may produce a large number of useless matches, which can flood useful information. Dual simulation [18] enhances graph simulation by imposing an additional condition, to preserve both child and parent relationships (downward and upward mappings). Due to the good balance and high practical value of dual simulation in response time and effectiveness, graph pattern matching is defined as dual simulation in this paper.

At present, the study of incremental graph pattern matching is still in its infancy; existing work [6–12] mainly focuses on the updates of data graphs. Fan et al. proposed the incremental graph simulation algorithm IncMatch [6, 7]. Sun et al. studied the Maximal Clique Enumeration problem on dynamic graph [8]. Stotz et al. studied incremental inexact subgraph isomorphic problem [9]. Wang and Chen proposed an incremental approximation graph matching algorithm, which transformed the approximate subgraph search into vector space relation detection [10]. When inserting or deleting on the data graph, the vectors of relevant nodes are modified and whether the new vectors still contain the vector of the pattern graph is rechecked. Choudhury et al. developed a fast matching system StreamWorks for dynamic graphs [11]. The system can real-time detect suspicious pattern graphs and early warn high-risk data transfer modes on constantly updated network graphs. Semertzidis and Pitoura proposed an approach to find the most durable matches of an input graph pattern on graphs that evolve over time [12]. In [13], an incremental graph matching algorithm was proposed for updates of pattern graphs.

In big data era [23], graph computing is widely used in different fields such as social networks [24], sensor networks [25, 26], internet-of-things [27, 28], and cellular networks [29]. Therefore, there is urgent demand for improving the performance of big graph processing, especially graph pattern matching.

3. Model and Definition

For graph pattern matching, pattern graphs and data graphs are directed graphs with labels. Each node in graphs has a unique label, which defines the attitude of the node (such as keywords, skills, class, name, and company).

Definition 1 (graph). A node-labeled directed graph (or simply a graph) is defined as $G = (V, E, L)$, where V is a finite set of nodes, $E \subseteq V \times V$ is a finite set of edges, and L is a function that map each node u in V to a label $L(u)$; that is, $L(u)$ is the attribute of u.

Definition 2 (graph pattern matching). Given a pattern graph $P = (V_p, E_p, L_p)$ and a data graph $G = (V, E, L)$, P matches Q if there is a binary relation $R \subseteq V_p \times V$, such that

(1) if $(u, v) \in R$, then $L_p(u) = L(v)$;

(2) $\forall u \in V_p$, there exists a node v in G such that $(u, v) \in R$ and (a) $\forall (u, u') \in E_p$, there exists an edge $(v, v') \in E$

such that $(u', v') \in R$; (b) $\forall(u'', u) \in E_p$, there exists an edge $(v'', v) \in E$ such that $(u'', v'') \in R$.

Condition (2)(a) ensures that the matching node v keeps the child relationship of u; condition (2)(b) ensures that v maintains the parent relationship of u.

For any P and G, there exists a unique maximum matching relation R_M. Graph pattern matching is to find R_M, and the result graph G_r is a subgraph of G that can represent R_M.

Considering a real-life example, a recruiter wants to find a professional software development team from social network. Figure 2(a) is the basic organization graph of a software development team. The team consists of the following staffs with identity: project manager (PM), database engineer (DB), software architecture (SA), business process analyst (BA), user interface designers (UD), software developer (SD), and software tester (ST). Each node in the graph represents a person, and the label of node means the identity of person. The edge from node A to node B means that B works well under the supervision of A. A social network is shown in Figure 2(b). In this example, R_M is {(DB, DB_1), (PM, PM_1), (SA, SA_1), (BA, BA_1), (UD, UD_1), (SD, SD_1), (SD, SD_2), (ST, ST_1), (ST, ST_2)}. Because BA_2 does not have a child matching UD and SA_2 does not have a parent matching DB, PM_2 does not keep the child relationship of PM. For the same reason, SD_3 (resp., ST_3) does not match SD (resp., ST).

Definition 3 (incremental graph pattern matching for pattern graph changing). Given a data graph G and a pattern graph P, the matching result in G for P is $M(P, G)$. Assuming that P changes ΔP, the new pattern graph is expressed as $P \oplus \Delta P$. As opposed to batch algorithms that recompute matches starting from scratch, an incremental graph matching algorithm aims to find changes of ΔM to $M(P, G)$ in response to ΔP such that $M(P \oplus \Delta P, G) = M(P, G) \oplus \Delta M$.

When ΔP is small, ΔM is usually small as well, and it is much less costly to compute than to recompute the entire set of matches. In other words, this suggests that we compute matches once on the entire graph via a batch-matching algorithm and then incrementally identify new matches in response to ΔP without paying the cost of the high complexity of graph pattern matching.

In order to get ΔM quickly, indexes can be prebuilt based on the selected data features of graphs to reduce the search space during incremental matching. The more indexes, the shorter the time to get ΔM and the larger the space to store indexes. For large-scale data graphs, both response time and storage cost are needed to be reduced. Considering the balance of storage cost and response time, in this paper, three kinds of sets generated in the process of graph matching are used as index. (1) First are candidate matching sets $cand(\cdot)$; for each node u in P, $cand(u)$ includes all the nodes in G which only have the same label with u. The nodes in $cand(\cdot)$ are called c-nodes. (2) The second are child matching sets $sim(\cdot)$; for each node u in P, $sim(u)$ includes all the nodes in G which preserve the child relationship of u. The nodes in $sim(\cdot)$ are called s-nodes. (3) The third are complete matching sets $mat(\cdot)$; for each node u in P, $mat(u)$ includes all the nodes in G which preserve both the child and parent relationship of u. The nodes in $mat(\cdot)$ are called m-nodes.

The symbols used in this paper are shown in Notions Section.

4. iDeltaP_IncGPM Algorithm

In this section, we propose the improved incremental graph pattern matching algorithm for pattern graph changing (ΔP).

4.1. The Idea of PGC_IncGPM Algorithm. The basic framework of PGC_IncGPM [13] is shown in Figure 3.

The graph pattern matching algorithm (GPMS) is first performed on the entire data graph G for the pattern graph P. It computes the matching result graph G_r and creates the index needed for subsequent incremental matching. ΔP may include edge insertions (E^+) and edge deletions (E^-). Incremental graph pattern matching algorithm PGC_IncGPM first calls the subalgorithm AddEdges for E^+ to get G_r' and $index'$ and then calls the subalgorithm SubEdges for E^- to get G_r'' and $index''$. G_r'' is the new matching result $M(P \oplus \Delta P, G)$, and $index''$ is the new index that can be used for subsequent incremental matching if the pattern graph changes again.

Edge insertions (resp., edge deletions) in ΔP are processed one by one by AddEdges (resp., SubEdges). For example, when deleting multiple edges from P, the processing of PGC_IncGPM is as follows.

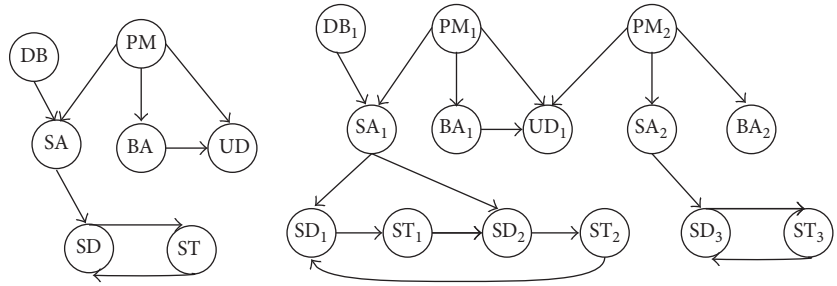
In the first step, the following operations are performed for each deleted edge (u, u'): for each $v \in cand(u)$, whether v keeps the child relationship of u in $P \oplus \Delta P$ is checked. If v keeps the child relationship of u, then v is removed from $cand(u)$ to $sim(u)$ and the parents of v in $cand(\cdot)$ are also processed.

In the second step, each node in $sim(\cdot)$ is repeatedly filtered according to its parents and children; the new generated m-nodes are added to $mat(\cdot)$.

In the first step, when deleting (u, u') from P, some nodes in $cand(u)$ and $cand(u'')$ (u'' is an ancestor of u) may change from c-nodes to s-nodes. So when a c-node becomes an s-node, a bottom-up approach is used to find its parents and ancestors from $cand(\cdot)$. If (u_1, u_1') and (u_2, u_2') are deleted, and u_1 and u_2 have a common ancestor u', then $cand(u')$ will be visited twice. In summary, there is a bottleneck of PGC_IncGPM for multiple deleted edges. There is the same problem for multiple inserted edges.

4.2. Optimization for Matching State Detection Sequence. Since PGC_IncGPM deals with edge insertions (resp., deletions) one by one, the efficiency of it gradually decreases as the number of changed edges increases. To overcome the bottleneck of PGC_IncGPM, multiple edge insertions (resp., deletions) should be considered together. In this paper, the optimization method for nodes' matching state detection sequence is proposed. The optimization can be applied to both insertions and deletions on P.

Taking SubEdges as an example, the optimization method is as follows.



(a) Pattern graph (b) Data graph

Figure 2: An example of graph pattern matching.

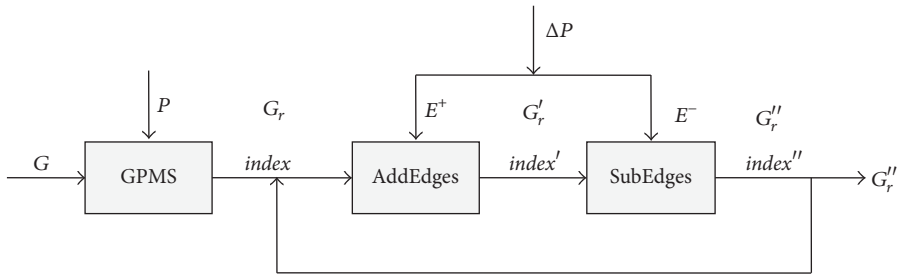


Figure 3: Basic framework of PGC_IncGPM algorithm.

First, analyze all edges deleted from P to determine which nodes' candidate matching sets may change. If $cand(u)$ may change, then u is added to $filtorder^-$ set.

Secondly, $filtorder^-$ is sorted by the inverse topological sequence of P. There may be some strong connected components in P. In this case, we first find out all the strong connected components in $P$and, then, converge each strong connected component into a node to get a directed acyclic graph P' and find the inverse topological sequence of P'; finally, we replace the strong connected component convergence node with the original node set. Thus, the approximate inverse topological sequence of P is obtained.

Finally, for each u in $filtorder^-$, $cand(u)$ is processed in turn. Depending on whether there is a deleted edge from u, two different filtering methods are used: (1) if u has at least one out-edge to be deleted, then each node in $cand(u)$ is likely to keep the child relationship of u now. So whether they keep the child relationship of u should be checked; (2) if u does not have an out-edge be deleted, then only part of the nodes in $cand(u)$ are needed to be checked. That is, a node in $cand(u)$ will be checked only if it has at least one child which changes from c-node to s-node.

The visited times of some candidate matching sets can be reduced through the above optimization.

4.3. iDeltaP_IncGPM Algorithm. Based on the optimization method proposed in Section 4.2, iDeltaP_IncGPM is proposed. It uses the optimized method for both multiple inserted edges and multiple deleted edges. The optimization algorithm for edge deletions is shown in Algorithm 1. In Algorithm 1, $nodes^-$ contains all the nodes which have out-edge deleted. For a node u in P, if the changes of P may result in some nodes in $cand(u)$ becoming s-nodes, then $u \in filtorder^-$. $Filtorder^-$ is sorted by the inverse topological sequence of P (lines (1)–(5)). If u has an out-edge removed, that is, $u \in nodes^-$, then all the nodes in $cand(u)$ need to be checked whether they keep the child relationship of u (lines (7)–(12)). If $u \in filtorder^-$ and u is not in $nodes^-$, then only part of nodes in $cand(u)$ are checked. That is, if w has a child w' and w' is moved from $cand(u')$ to $sim(u')$ ($w' \in snew(u')$), then whether w is still an s-node will be checked (lines (14)–(20)).

Here we use an example to illustrate the implementation process of PGC_IncGPM and iDeltaP_IncGPM. The pattern graph P is shown in Figure 4, assuming that (E, H), (G, I), and (C, G) are deleted from P.

The process of PGC_IncGPM is as follows. (1) the deletion of (E, H) is processed, and each w in $cand$(E) is checked whether it keeps the child relationship of u in $P \oplus \Delta P$. If w keeps the child relationship of u, then its parents founded from $cand$(B) (resp., $cand$(C)) are checked. If these nodes keep the child relationship of B (resp., C), then they are removed to sim(B) (resp., sim(C)). After that, their parents founded from $cand$(A) are checked; (2) the deletion of (G, I) is processed, and the nodes in $cand$(G), $cand$(C), $cand$(D), and $cand$(A) are checked in turn; (3) the deletion of (C, G) is processed, and the nodes in $cand$(C) and $cand$(A) are checked in turn. From the above steps, it can be seen that $cand$(C) and $cand$(A) are visited three times, $cand$(G), $cand$(D), $cand$(E), and $cand$(B) are visited once.

The process of iDeltaP_IncGPM is as follows: because of the deletion of (E, H), (G, I), and (C, G), some nodes in $cand$(E), $cand$(G), $cand$(C), $cand$(B), $cand$(D), and $cand$(A) may become s-nodes. The nodes in $cand(\cdot)$ are checked by

```
(1)  nodes⁻ = Φ; filtorder⁻ = Φ;
(2)  for each deleted edge e = (u, u′) do
(3)    nodes⁻ = nodes⁻ ∪ {u};
(4)    filtorder⁻ = filtorder⁻ ∪ {u and all ancestor nodes of u};
(5)   sort filtorder⁻ according to the inverse topological of P;
(6)  for each node u in filtorder⁻ do
(7)    if u ∈ nodes⁻ then
(8)      for each w ∈ cand(u) do
(9)        check if w keeps the child relationship of u;
(10)       if w keeps the child relationship of u then
(11)         sim(u) = sim(u) ∪ {w}; cand(u) = cand(u)\{w};
(12)         snew(u) = snew(u) ∪ {w};
(13)   else
(14)     for each w ∈ cand(u) do
(15)       if there exist w′ which is a child of w such that (u, u′)∈ E_p, w′ ∈ snew(u′) then
(16)         check if w keeps the child relationship of u;
(17)         if w keeps the child relationship of u then
(18)           sim(u) = sim(u) ∪ {w};
(19)           cand(u) = cand(u)\{w};
(20)           snew(u) = snew(u) ∪ {w};
(21) repeatly filter sim(·) according to the parent and child relationships of nodes in the
       subgraph constructed by sim(·) to get added mat(·) and updated sim(·) and cand(·);
```

Algorithm 1: iDeltaP_IncGPM for edge deletions.

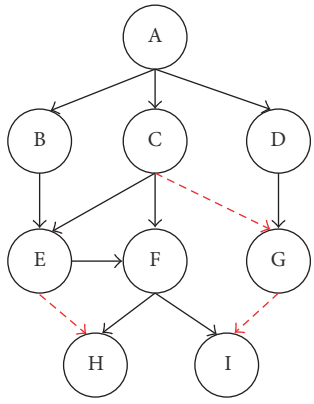


Figure 4: An example for pattern graph changing.

the order {G, E, D, C, B, A}. That is, the nodes in *cand*(G) are checked first, and the nodes in *cand*(A) are checked at last. E, G, and C all have out-edges deleted, so all the nodes in their candidate matching sets are checked. For the nodes in *cand*(B), *cand*(D), and *cand*(A), only if they have a child changing from *c*-node to *s*-node, they will be checked. Therefore, *cand*(C), *cand*(D), *cand*(A), *cand*(G), *cand*(E), and *cand*(B) are only visited once. In other words, the optimized scheme reduces the visited times of *cand*(·).

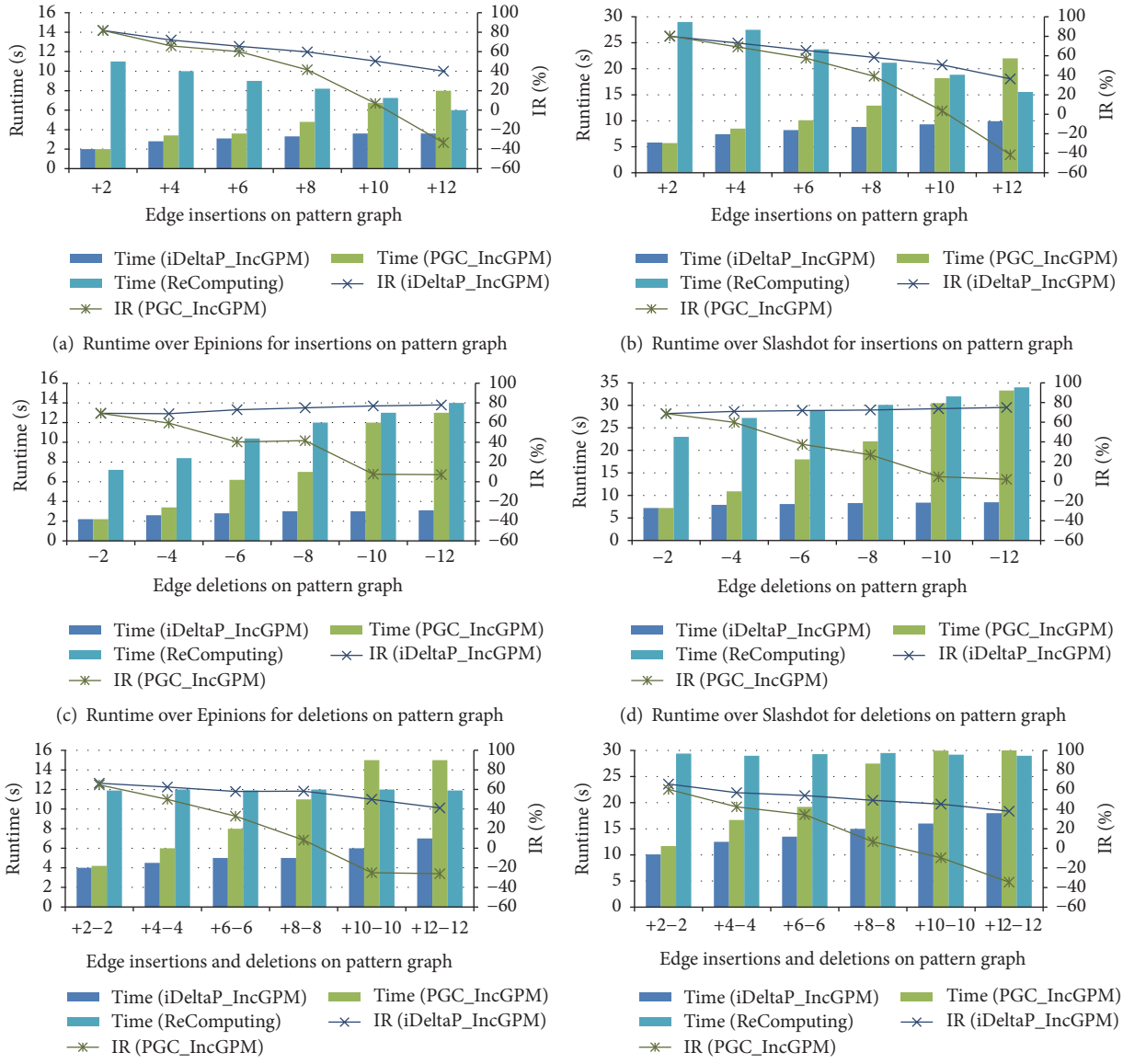
For multiple edges inserted to the pattern graph, the similar optimization method is adopted. $nodes^+$ contains all the source nodes of inserted edges. If some nodes in $sim(u)$ may become *c*-nodes because of edge insertions, then u is in $filtorder^+$. $filtorder^+$ is ordered by the reverse topological sequence of the pattern graph. $nodes^+$ and $filtorder^+$ are used to reduce the visited times of $sim(\cdot)$ and $mat(\cdot)$.

5. Experiments and Results Analysis

The following experiments evaluate our proposed algorithm. Runtime is used as a key assessment of algorithms. In addition, in order to show the effectiveness of incremental algorithms visually, improvement ratio (IR) is proposed, which is the ratio of runtime saved by incremental matching algorithms to the runtime of ReComputing algorithm. Two real data sets (Epinions and Slashdot [30]) are used for experiments. The former is a trust network with 75879 nodes and 508837 edges. The latter is a social network with 82168 nodes and 948464 edges. In previous work, we experimented with normal size and large size pattern graphs, respectively, and the results show that the complexity and effectiveness of incremental matching algorithm are not affected by the size of pattern graph. Therefore, in this paper, by default, the number of nodes in P ($|V_p|$) is 9, the original number of edges in P ($|E_p|$) is 8 (resp., 16) for insertions (resp., deletions) and 9 for both insertions and deletions.

In order to evaluate the improvement of our proposed algorithm, iDeltaP_IncGPM, PGC_IncGPM, and ReComputing are all performed on Epinions and Slashdot under different settings. Each experiment was performed 5 times with different pattern graphs, and the average results are reported here. The experimental results are shown in Figure 5. The histogram represents the runtime of algorithm, and the line chart represents the improvement ratio of iDeltaP_IncGPM and PGC_IncGPM to ReComputing.

Figure 5(a) (resp., Figure 5(b)) shows the runtime of three algorithms over Epionions (resp., Slashdot) for insertions on pattern graphs. The X-axis represents the number of insertions on P, "+2" represents that two edges are inserted to P, "+4" represents four edges are inserted



(a) Runtime over Epinions for insertions on pattern graph

(b) Runtime over Slashdot for insertions on pattern graph

(c) Runtime over Epinions for deletions on pattern graph

(d) Runtime over Slashdot for deletions on pattern graph

(e) Runtime over Epinions for both insertions and deletions on pattern graph

(f) Runtime over Slashdot for both insertions and deletions on pattern graph

Figure 5: The runtime of different algorithms when pattern graph changed.

to P, and so on. The figure tells us the following: (a) when insertions are no more than 10, the runtime of PGC_IncGPM and iDeltaP_IncGPM is significantly shorter than that of ReComputing, and iDeltaP_IncGPM has the shortest runtime; (b) when insertions are 12 (new inserted edges account for 60% of the edges in $P \oplus \Delta P$), the runtime of PGC_IncGPM is longer than that of ReComputing, while iDeltaP_IncGPM still gets the shortest runtime; (c) the improvement ratio of iDeltaP_IncGPM and PGC_IncGPM decreases with the increase of edge insertion, but the decrease of iDeltaP_IncGPM is smaller. The more inserted edges, the better iDeltaP_IncGPM than PGC_IncGPM. When 12 edges are inserted to P, the IR of iDeltaP_IncGPM is 40% on average, and the IR of PGC_IncGPM is 33% on average. Therefore, iDeltaP_IncGPM is better than PGC_IncGPM. The reason is that PGC_IncGPM processes the inserted edges one by one. Therefore, as insertion increases, its runtime grows almost linearly. However, iDeltaP_IncGPM integrates all the inserted edges, analyzes which matching sets are affected, and processes them in the appropriate order. This will prevent some matching sets to be processed repeatedly, which will shorten the running time.

Figure 5(c) (resp., Figure 5(d)) shows the runtime of three algorithms over Epionions (resp., Slashdot) for deletions on pattern graph. The X-axis represents the number of deletions on P, "−2" represents that two edges are deleted from P, "−4" represents four edges are deleted from P, and so on. It can be seen that (a) when deletion changes from 2 to 12, the runtime of all three algorithms increases, and iDeltaP_IncGPM always has the shortest runtime; (b) as the deletion increases, the IR

of PGC_IncGPM decreases and the IR of iDeltaP_IncGPM slowly increases. For 12 deletions, the IR of PGC_IncGPM decreases to 7% on average, while the IR of iDeltaP_IncGPM increases to 78% on average. The reason is that as the deletion increases, the runtime of ReComputing increases dramatically, while the runtime of iDeltaP_IncGPM increases a little. iDeltaP_IncGPM is better than PGC_IncGPM because it compositely processes deleted edges and its runtime does not increase linearly as the number of deleted edges increases.

Figure 5(e) (resp., Figure 5(f)) shows the runtime of three algorithms over Epionions (resp., Slashdot) for both insertions and deletions on pattern graph. The X-axis represents the number of insertions and deletions on P, "+2−2" means that two edges are inserted to P and the other two edges are removed from P, and so on. As shown in the figure, iDeltaP_IncGPM always has shorter runtime than the others do.

In conclusion, iDeltaP_IncGPM effectively improves the efficiency of PGC_IncGPM through the optimization strategy. For the same ΔP, the runtime of iDeltaP_IncGPM is shorter, and as $|\Delta P|$ increases, the runtime increases less; the decrease of IR is also more moderate. Therefore, iDeltaP_IncGPM can be applied to larger changes of the pattern graph, and it has a wider range of applications.

6. Conclusion

In this paper, we analyze PGC_IncGPM to find its efficiency bottleneck and propose a more efficient incremental matching algorithm iDeltaP_IncGPM. Multiple insertions (resp., deletions) are considered together and the optimization method for nodes' matching state detection sequence is used. Experimental results on real data sets show that iDeltaP_IncGPM has higher efficiency and wider application range than PGC_IncGPM.

Next, we will study the distributed incremental graph matching algorithm. Real-life graphs grow rapidly in size and hyper-massive data graphs cannot be centrally stored in one data center and need to be distributed across multiple data centers. It is very worthy studying how to make efficient incremental matching on distributed large graphs.

Notations

P/G:	Pattern graph/data graph
u/u':	Nodes in P
v/v':	Nodes in G
ΔP:	Changes of P
ΔG:	Changes of G
$P \oplus \Delta P$:	New pattern graph
$cand(u)$:	Nodes in G that have same label with u but do not keep child relationship of u
$sim(u)$:	Nodes in G that only keep child relationship of u
$mat(u)$:	Nodes in G that keep child and parent relationships of u
index:	The sets including $cand(u)$, $sim(u)$ and $mat(u)$
$c/s/m\text{-}node$:	v in G such that $v \in cand(u)/sim(u)/mat(u)$
$M(P, G)$:	The maximum match in G for P
Gr:	The result graph, a subgraph represents $M(P, G)$.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] X. Ren and J. Wang, "Multi-query optimization for subgraph isomorphism search," *Proceedings of the VLDB Endowment*, vol. 10, no. 3, pp. 121–132, 2016.

[2] Z. Yang, A. W.-C. Fu, and R. Liu, "Diversified top-k subgraph querying in a large graph," in *Proceedings of the 2016 ACM SIGMOD International Conference on Management of Data, SIGMOD 2016*, pp. 1167–1182, USA, July 2016.

[3] J. Gao, B. Song, W. Ke, and X. Hu, "BalanceAli: Multiple PPI Network Alignment With Balanced High Coverage and Consistency," *IEEE Transactions on NanoBioscience*, vol. 16, no. 5, pp. 333–340, 2017.

[4] A. Jentzsch, "Linked Open Data Cloud," in *Linked Enterprise Data*, X.media.press, pp. 209–219, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[5] Y. Hao, G. Li, P. Yuan, H. Jin, and X. Ding, "An Association-Oriented Partitioning Approach for Streaming Graph Query," *Scientific Programming*, vol. 2017, pp. 1–11, 2017.

[6] W. Fan, J. Li, J. Luo, Z. Tan, X. Wang, and Y. Wu, "Incremental graph pattern matching," in *Proceedings of ACM SIGMOD and 30th PODS 2011 Conference*, pp. 925–936, Greece, June 2011.

[7] W. Fan, C. Hu, and C. Tian, "Incremental Graph Computations," in *Proceedings of ACM International Conference*, pp. 155–169, Chicago, Illinois, USA, May 2017.

[8] S. Sun, Y. Wang, W. Liao, and W. Wang, "Mining Maximal Cliques on Dynamic Graphs Efficiently by Local Strategies," in *Proceedings of IEEE 33rd International Conference on Data Engineering (ICDE)*, pp. 115–118, San Diego, CA, USA, April 2017.

[9] A. Stotz, R. Nagi, and M. Sudit, "Incremental graph matching for situation awareness," in *Proceedings of 12th International Conference on Information Fusion, FUSION 2009*, pp. 452–459, usa, July 2009.

[10] C. Wang and L. Chen, "Continuous subgraph pattern search over graph streams," in *Proceedings of the 25th IEEE International Conference on Data Engineering, ICDE 2009*, pp. 393–404, China, April 2009.

[11] S. Choudhury, L. Holder, G. Chin, A. Ray, S. Beus, and J. Feo, "Streamworks - A system for dynamic graph search," in *Proceedings of ACM SIGMOD Conference on Management of Data, SIGMOD 2013*, pp. 1101–1104, USA, June 2013.

[12] K. Semertzidis and E. Pitoura, "Durable Graph Pattern Queries on Historical Graphs," in *Proceedings of International Conference on Data Engineering*, pp. 541–552, October 2016.

[13] L. X. Zhang, W. P. Wang, J. L. Gao, and J. X. Wang, "Pattern graph change oriented incremental graph pattern matching," *Journal of Software. Ruanjian Xuebao*, vol. 26, no. 11, pp. 2964–2980, 2015.

[14] X. Ren and J. Wang, "Exploiting Vertex Relationships in Speeding up Subgraph Isomorphism over Large Graphs," in *Proceedings of the 3rd Workshop on Spatio-Temporal Database Management, Co-located with the 32nd International Conference on Very Large Data Bases, VLDB 2006*, pp. 617–628, kor, September 2006.

[15] F. Bi, L. Chang, X. Lin, L. Qin, and W. Zhang, "Efficient subgraph matching by postponing Cartesian products," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 1199–1214, USA, July 2016.

[16] J. R. Ullmann, "An algorithm for subgraph isomorphism," *Journal of the ACM*, vol. 23, no. 1, pp. 31–42, 1976.

[17] W. Fan, J. Li, S. Ma, N. Tang, Y. Wu, and Y. Wu, "Graph pattern matching," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 264–275, 2010.

[18] W. Fan, "Graph pattern matching revised for social network analysis," in *Proceedings of the 15th International Conference on Database Theory, ICDT 2012*, pp. 8–21, deu, March 2012.

[19] J. Gao, Q. Ping, and J. Wang, "Resisting re-identification mining on social graph data," *World Wide Web-Internet and Web Information Systems*, 2017.

[20] S. Ma, Y. Cao, W. Fan, J. Huai, and T. Wo, "Capturing topology in graph pattern matching," *Proceedings of the VLDB Endowment*, vol. 5, no. 4, pp. 310–321, 2011.

[21] A. Fard, M. U. Nisar, L. Ramaswamy, J. A. Miller, and M. Saltz, "A distributed vertex-centric approach for pattern matching in massive graphs," in *Proceedings of IEEE International Conference on Big Data, Big Data 2013*, pp. 403–411, USA, October 2013.

[22] Y. Liang and P. Zhao, "Similarity Search in Graph Databases: A Multi-Layered Indexing Approach," in *Proceedings of IEEE 33rd International Conference on Data Engineering (ICDE)*, pp. 783–794, San Diego, CA, USA, April 2017.

[23] X. Liu, Y. Liu, H. Song, and A. Liu, "Big Data Orchestration as a Service Network," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 94–101, 2017.

[24] J. Gao, J. Wang, J. He, and F. Yan, "Against Signed Graph Deanonymization Attacks on Social Networks," *International Journal of Parallel Programming*.

[25] Q. Zhang and A. Liu, "An unequal redundancy level-based mechanism for reliable data collection in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, article 258, 2016.

[26] J. Gao, J. Wang, P. Zhong, and H. Wang, "On Threshold-Free Error Detection for Industrial Wireless Sensor Networks," *IEEE Transactions on Industrial Informatics*, pp. 1–11.

[27] Y. Xu, A. Liu, and C. Changqin, "Delay-aware program codes dissemination scheme in internet of everything, mobile information systems," *Mobile Information Systems*, vol. 2016, Article ID 2436074, 18 pages, 2016.

[28] X. Liu, S. Zhao, A. Liu, N. Xiong, and A. V. Vasilakos, "Knowledge-aware Proactive Nodes Selection approach for energy management in Internet of Things," *Future Generation Computer Systems*, 2017.

[29] K. Zhou, J. Gui, and N. Xiong, "Improving cellular downlink throughput by multi-hop relay-assisted outband D2D communications," *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, 2017.

[30] Stanford Large Network Dataset Collection, http://snap.stanford.edu/data/index.html.

Hindawi
Scientific Programming
Volume 2017, Article ID 2056501, 7 pages
https://doi.org/10.1155/2017/2056501

Research Article

Routing Optimization Algorithms Based on Node Compression in Big Data Environment

Lifeng Yang,[1] Liangming Chen,[2] Ningwei Wang,[2] and Zhifang Liao[2]

[1]*School of Continuing Education, Yunnan Open University, Yunnan, China*
[2]*School of Software, Central South University, Hunan, China*

Correspondence should be addressed to Zhifang Liao; zfliao@csu.edu.cn

Received 26 August 2017; Accepted 5 December 2017; Published 26 December 2017

Academic Editor: Wenbing Zhao

Shortest path problem has been a classic issue. Even more so difficulties remain involving large data environment. Current research on shortest path problem mainly focuses on seeking the shortest path from a starting point to the destination, with both vertices already given; but the researches of shortest path on a limited time and limited nodes passing through are few, yet such problem could not be more common in real life. In this paper we propose several time-dependent optimization algorithms for this problem. In regard to traditional backtracking and different node compression methods, we first propose an improved backtracking algorithm for one condition in big data environment and three types of optimization algorithms based on node compression involving large data, in order to realize the path selection from the starting point through a given set of nodes to reach the end within a limited time. Consequently, problems involving different data volume and complexity of network structure can be solved with the appropriate algorithm adopted.

1. Introduction

The single source shortest path problems in graph theory are very typical questions that enjoy wide applications in real life, such as network routing path selection, vehicle navigation, and travel routes. The classic algorithm to solve such problems is Dijkstra's Algorithm [1] proposed by Dijkstra in 1959 and a lot of researchers focus on this research area [2–4]. However, Dijkstra fails to solve problems where routes are required to go from the starting point, pass the specified intermediate node, and finally reach the destination—far more practical problems exemplified as follows:

(1) "Postman problem": the postman starts from the post office, sends letters to residents, and returns home, where we need to find the postman a shortest path within a given time.

(2) "Limited time problem": within a limited time, activities designed for staff members who tracked consent using depth sensors were proposed and they were carefully reminded of noncompliant activities [5], and a collaborative smartphone task model is proposed, which is called Collaboration-Based Intelligent Perception Task Model (CMST) [6].

(3) "Traveler problem": calculate a travel route for the traveler within the specified time, who needs to go from a designated location, pass a designated scenery spot, and visit a given place. The total distance should be the shortest or the total expense should be the lowest [7, 8].

(4) "Compression problem": a new compression method for large data environment is proposed, which can effectively reduce the data compression of single nodes and ensure the quality of data [9]. Due to the large amount of web service data, a data-driven scheme is based on kernel least mean squares (KLMS) algorithm [10]. In order to compress the input to further improve the learning effect, a new QKLMS is based on entropy-guided learning [11].

(5) "Network routing problem": find an efficient routing algorithm to solve the problem of path optimization of wireless sensor network, considering the influences of some practical factors such as the consumption of the energy of the nodes and recovery time of routing [12–14].

(6) "Laguerre neural network" [15]: it intends to propose a novel automatic learning scheme to improve the tracking efficiency while maintaining or improving the data tracking accuracy. A core strategy in the proposed scheme is the design

of Laguerre neural network- (LaNN-) based approximate dynamic programming (ADP).

(7) "Energy of the sensor nodes" [16]: a novel prediction-based data fusion scheme using grey model (GM) and optimally pruned extreme learning machine (OP-ELM) is proposed. The proposed data fusion scheme called GM-OP-ELM uses a dual prediction mechanism to keep the prediction data series at the sink node and sensor node synchronous.

These problems can be summarized as one graph theory problem; that is, in a weighted directed graph, a route goes from a starting point, passes through the designated intermediate node, and reaches a destination. It is required to find valid paths within a specified time, calculate the weight of these paths, and select a path with the lowest weight as the final result.

To solve this kind of problems, we may traverse the whole graph and find a shortest path, although theoretically this traversal algorithm will eventually sort out the optimal solution; however the time complexity remains high. In view of this, this paper proposes a node compression routing algorithm with considered time limits. The study pays attention to node compression and applies useful information obtained in path finding to search conditions, readjusting the order of subnodes and other methods as well. Additionally, the high time complexity in traditional algorithm is improved, offering an effective solution to this type of problem.

2. Problem Description

2.1. Mathematical Model of the Problem. Given a weighted graph $G(V, E)$ where $V = \{1, 2, 3, \ldots, n\}$ is the vertex set, $E = \{e_{ij} = (i, j) \mid i, j \in V,\ i \neq j\}$ is the edge set. d_{ij} $(i, j \in V,\ i \neq j)$ is the weight of vertexes i to j, where $d_{ij} > 0$ and $d_{ij} \neq \infty$; while d_{ij} and d_{ji} may be unequal, $V' = \{1', 2', \ldots, n'\} \in V$. We need to find the sequence $A = \{a_1, a_2, a_3, \ldots, a_n\}$ within a given time, where s is starting point and t is the destination, $s, t \in V$ and s, t do not belong to V', all of the elements in V' must appear in sequence A, making the sum of the weights of all edges of the path formed in sequence A minimal, and loop is not allowed in any path. The mathematical model of the problem is defined as follows.

Under the condition of Time $= t$, solve $\min C = \sum_{i \neq j} d_{ij} \times X_{ij}$, in order to define the starting point s and the destination t and make sure that there's only one in-edge and out-edge on each vertex except the edges of starting point and the destination paths; we make the following constraints:

$$X_{ij} = \begin{cases} 1, & \text{edge } e_{ij} \text{ is along the result path} \\ 0, & \text{edge } e_{ij} \text{ is out of the result path,} \end{cases} \quad (1)$$

where X_{ij} is an integer of 0 or 1, 1 represents edge e_{ij} on the result path, and 0 represents edge e_{ij} out of the result path, and X_{ij} is used to calculate the weight of the resulting path.

$$\sum_{i \neq j} X_{ij} = 1, \quad j \in V', \quad (2)$$

where $i \neq j$ means that the result path cannot contain the edges that the starting node and the end node are the same node, which means the point in the intermediate node set on the result path can only occur once and must occur once.

$$\sum X_{sj} = 1, \quad j \in V,\ j \neq s. \quad (3)$$

The formula defines an edge that begins with the starting nodes which should appear in the result path, and the starting node in the edge cannot be the end node.

$$\sum X_{js} = 0, \quad j \in V,\ j \neq s. \quad (4)$$

The formula restricts that the starting node s can only be the starting node in an edge, and it cannot be any other kind of nodes, such as end node or intermediate nodes.

$$\sum X_{it} = 1, \quad i \in V,\ i \neq t. \quad (5)$$

The formula restricts that the result path must have an edge ended with the end node t, which means the edge cannot start with the end point t.

$$\sum X_{ti} = 0, \quad i \in V,\ i \neq t. \quad (6)$$

The formula restricts that the resulting path cannot contain the edge beginning with the end node t; that is, the end node t can only be used as the final node on the resulting path.

$$\sum_{i,j \in V} X_{ij} = |A|. \quad (7)$$

This formula defines the number of edges on the resulting path which can be the number of nodes minus one; that is, the resulting path cannot appear with unrelated edges and loops.

For the convenience of subsequent description, the following two definitions are given.

Definition 1 (key nodes). The nodes in V' include other must-pass nodes except starting point s and destination t.

Definition 2 (free nodes). All other nodes except the key nodes are included.

2.2. Simple Example. In the weighted graph G shown in Figure 1, four nodes can be found, namely, 0, 1, 2, and 3; therefore $V = \{0, 1, 2, 3\}$, and there are seven edges 0, 1, 2, 3, 4, 5, and 6, so $E = \{0, 1, 2, 3, 4, 5, 6\}$, where the weight of the edge is $\{d_{01} = 1,\ d_{02} = 2,\ d_{03} = 1,\ d_{21} = 3,\ d_{31} = 1,\ d_{23} = 1,\ d_{32} = 1\}$. To find a path from 0 to 1 via vertexes 2 and 3, we have $V' = \{2, 3\}$. Two paths can be found to solve this problem: 0→2→3→1 and 0→3→2→1. Since the weight of edges on the first route is 4, and the weight of the other is 5, the optimal solution should be 0→2→3→1.

3. Improved Backtracking Algorithm: IBA

If using the backtracking method to solve this problem, theoretically, we can have the optimal solution and of course other solutions. However, the backtracking method does not effectively use information constructed in the search process or the optimal solution to lay a foundation for optimization condition of the next-step search. In this section, an

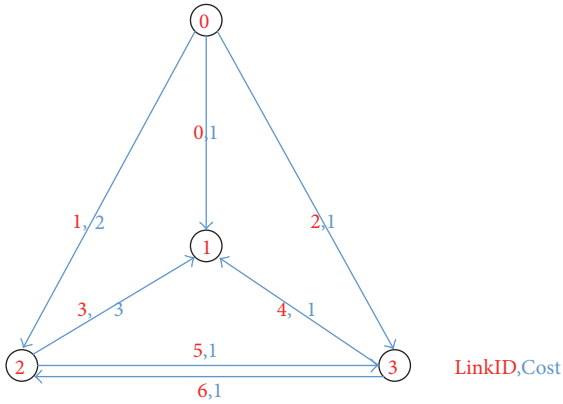


Figure 1: A simple example of the problem.

```
Improved-Backtrack (G)
(1) node=start
(2) while usedtime< t &&
(node!=end && !A′ ∈ nodes)
(3) nodes.add(node)
(4) record information include
route and weigths
(5) for i = 1 to children.length
(6) add search rule
(7) Improvedacktrack (children[i])
(8) if result !=null-B
(9) return result and weight
(10) else
(11) return NA
```

Algorithm 1: The key pseudocode of the improved backtracking algorithm.

improved backtracking method (OPT-Backtrack Algorithm) is proposed based on traditional backtracking method. The new IBA retrieves known information and valid results from the previous search and adds them up to the next search rules before searching from other nodes. In this way, the search method and algorithms can be improved, since existing information and possible results are taken into consideration for a higher search efficiency.

The addition rule in the improved backtracking algorithm is shown below.

Rule 1. If the next node happens to be the destination, yet the current path has not gone through every must-pass node in the node set, the path will track back and begin searching for the next node. This rule avoids the generation of many invalid solutions thus improving the algorithm efficiency.

Rule 2. If the current path weight and the weight of the edge to the next node is greater than or equal to the minimum weight of the available solution, the path will track back and continue searching for the next node. If current path has been found whose current weight and the weight of edge to the next node is no more than the existing weight, then there is no need to search for the next node, because initially the problem is to find the smallest possible weight of the path.

Rule 3. For those nondestination nodes with zero child nodes, we should avoid entering the search. If a node is not destination and has no child nodes, the path shall not continue; therefore, it is not necessary to search at such nodes or rather they can be simply deleted from the graph.

The key pseudocode of the improved backtracking algorithm is shown in Algorithm 1.

4. Node Compression Based Search Algorithm

Although search efficiency can be enhanced by the improved backtracking algorithm to a certain degree, the negative complexity of the improved backtracking method will also increase as scale of the graph and solution domain expand. To reduce algorithm complexity, this paper proposes a new algorithm, node compression based search algorithm: NCSA.

As the scale of graph increases, paths will expand accordingly. The same problem would be finding a path from a start point, reaching an intermediate node halfway and finally the destination. To reduce the algorithm complexity, we may preprocess the graph. The method is to compress the total number of nodes, remove useless nodes and low-value path fragments, and then save the only paths that are necessary to simplify the entire graph; the goal is to compress solution domain and ultimately improve search efficiency.

4.1. Node Compression Algorithm (NCA). The algorithm is applicable to the following circumstance: If a node is relatively remote which only reaches one other node, that is, a node followed only by one child node, in this case, the search will follow down the only child node route and will repeat this wherever there is such a node during the searching process. What we want to do is to avoid the simple and repeated calculations in this kind of situation.

Solution to this problem is Node Compression Algorithm (NCA). NCA records the paths through the above-mentioned nodes when the algorithm is applied for the first time and will remove the nodes but retain the path information; therefore, when the next search continues at this node, only stored path information will be used to avoid duplicated counting. As a result, the total number of nodes is compressed and reduced, making it easier to search for a better solution.

The process is shown in Figure 2.

In Figure 2, node 1 is followed by the only child node 2, the weight from nodes 1 to 2 is 2, marked as path 1; the compression process means transferring node 1 information to node 2 so that node 2 becomes the direct child node of node 0. If compressed, the weight from nodes 0 to 2 is 3, and path from nodes 0 to 2 is "0 | 1." This means node 1 is removed while the path information from nodes 1 to 2 is retained solely in node 2. When the next search algorithm reaches node 0, information retained in node 2 can be used directly without going back to node 1. So the number of nodes is reduced and the path will not be searched again.

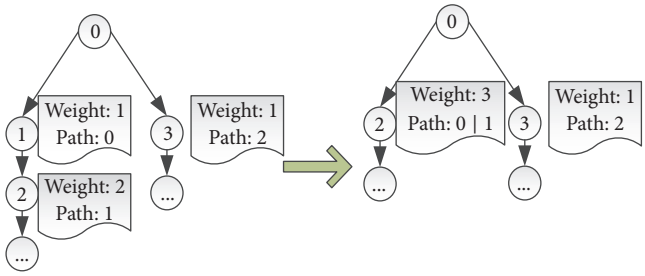


Figure 2: The basic idea of compression search algorithm.

4.2. Complete Compression Algorithm: CCA. Since Node Compression Algorithm (NCA) is used mainly to solve free nodes with only one child node, if such nodes are many in the graph, the algorithm efficiency will be significantly improved. However, if the scale of such nodes is limited, the basic compression algorithm will take less or no effect, which limits the effectiveness of compression search algorithm.

In view of the problem of NCA, this paper proposes a more efficient compression strategy, which compresses all free nodes in the graph to reduce the complexity of the graph, improving the search efficiency.

The problem is finding a noncircle path from the starting node to the destination node while passing through the intermediate node sets so that the weights of the edges on paths are as small as possible. When the reachability of nodes is complex, there will be many more possible paths to reach nodes of one and another. Since the problem requires that intermediate node set V' be passed and, within the set, there are multiple reachable paths between nodes, yet only one path will be selected within the set as one fragment of the final solution, therefore, we should find out all reachable paths while saving the path with the smallest weight. As the search algorithm reaches a corresponding node, the valid path will be retrieved from the stored information while the original nodes on the path can be removed from the graph, reducing useless nodes and repetitive counting. With this compression method, only the starting point, destination, the intermediate node set, and their interconnected path information will remain, simplifying the entire graph to a large extent with excellent compression efficiency.

Just like Figure 1, it can be seen as a simplified graph, and only the starting point, destination, and intermediate node set are preserved. In this way, we can achieve good compression efficiency by selecting the reachable path with the smallest path.

4.3. Improved Complete Compression Algorithm: ICCA. In order to further improve compression efficiency, this section continues to adjust and improve node compression by the three steps.

4.3.1. Adjusting Child Nodes Order by Weight. In the search process, algorithm can be done based on the weight of feasible solutions (see Rule 2 of IBA). First the order of subnodes is sorted according to the weight size from small to large. When algorithm searches the path, subnodes carrying smaller weight are searched with priority so that paths with smaller weight are easily obtained. As a result of this search strategy, other paths with larger weight can be skipped. This certainly reduces unnecessary search processes with greater efficiency.

4.3.2. Adjusting Child Nodes Order by the Sequence of Passing Nodes (from Small to Large). From the perspective of probability, when a new node is inserted into a graph, the more the nodes a path passes, the more likely the repeated path will be generated. Therefore, under the condition of same weight, the nodes with fewer subnodes will be given priority since the paths that follow will make fewer repeated attempts, making it easier to find the solution path.

4.3.3. Removing Child Nodes with Larger Weight. This strategy is only applicable to high-complexity graphs. After compression, the remaining nodes will connect one and another to form paths; complexity of the graph might be still high. There would be the case where one path might be an effective solution but the nodes it passes carry excessive weight, so the path will not be considered the final solution. In this case, removing large weight nodes will lower the graph complexity and improve search efficiency. In addition, it will save time and figure out a better solution with a lower weight path.

By analysis, the spatial complexity of IBA is $O(n)$, while the spatial complexity of NCA, CCA, and ICCA is $O(n^2)$, where n is the total number of nodes in the graph. ICCA can quickly select the shortest paths according to the weights of nodes and the nodes with smaller weights and delete the nodes with larger weights from the compression of large networks efficiently.

5. Experimental Analysis

5.1. Data Description and Analysis. Without loss of generality, experiment data are from the cases of *2016 Huawei Software Elite Competition*; these quoted examples are based on the network topological graph of Huawei's network routers, switches, and other network elements when Huawei established its own network facilities.

5.1.1. Problem Description. Given a weighted graph $G = (V, E)$, V is the vertex set, E is the directed edge set, and each directed edge contains the weight. For a given vertex s, t, and a subset V' of V, find a nonringing directed path P from s to t within a given time so that P passes through all vertices in V' (the order of passing is not required), making the total weight of all directed edges on path P as small as possible.

5.1.2. Data Description. (1) All weights in the graph are integers within $[1, 20]$.

(2) The starting point of any directed edge is not destination.

(3) The number of directed edges connecting vertex A to vertex B may be more than one, whose weight may or may not be the same.

(4) The total number of vertices of the directed graph will not exceed 600, and the number of each vertex out-degree

(the number of directed edges with these points as the starting point) does not exceed 8.

(5) The number of elements in V' does not exceed 50.

(6) The nonringing directed path P starts from s to t, where P is a directed connected path consisting of a series of directed edges from s to t, with no repeated path allowed.

(7) The weight of a path is the sum of all weights on the directed edges of the path.

5.1.3. Data Format. (1) In the graph, each line contains the following information:

{LinkID, SourceID, DestinationID, Cost},

where LinkID is index of directed edge, SourceID is index of the starting vertex of the directed edge, DestinationID is the index of destination vertex of the directed edge, Cost is the weight of the directed edge. The index of vertex and that of directed edge are numbered from 0 (not necessarily continuous, but the case ensures that the index does not repeat).

(2) Path information includes

{SourceID, DestinationID, IncludingSet},

where SourceID is the starting point of the path, DestinationID is the destination of the path, and IncludingSet represents the must-pass vertex set V', and different vertex indexes are segmented with "|."

5.1.4. Experiment Environment. Windows 7 64-bit operating system, with Intel core i5 processor, jre1.6, 32-bit java virtual machine, up to 4 G memory, is used.

5.2. Experiment Methods and Result Analysis

5.2.1. IBA, NCA, and CCA Comparison. To verify backtracking method and IBA, NCA, and CCA algorithms, four sets of experiments will be conducted with the solution time limited to 10 seconds. From Experiments 1–4, the total number of nodes and edges in the graph will be gradually increased, while the number of intermediate nodes will be kept unchanged. Experiment results will be compared by the weight of final path result and time spent.

Experiment 1. Total nodes are 10; must-pass nodes are 3; edges are 39.

Figure 3 shows the experimental result from Experiment 1 and it presents the fact that IBA has higher efficiency than the backtracking method. Efficiency difference is not remarkably obvious in NCA and CCA because the compression process takes time and also the efficiency becomes even less obvious if the complexity of the graph is low.

Experiment 2. Total nodes are 20; must-pass nodes are 5; edges are 55.

Figure 4 shows the experimental result from Experiment 2 and it presents the fact that IBA, NCA, and CCA have a greater efficiency than backtracking method. Efficiency

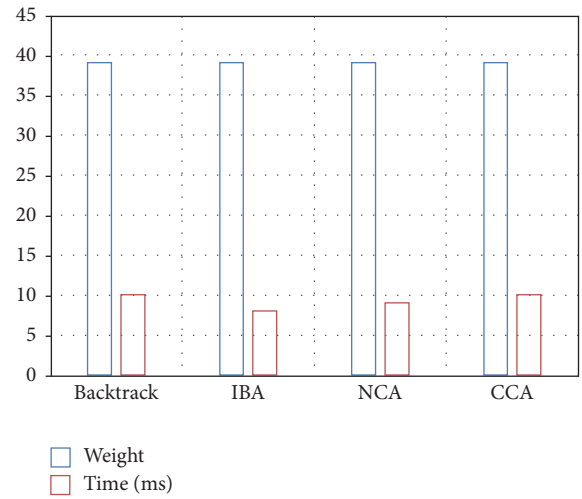


Figure 3: Experimental results of Experiment 1.

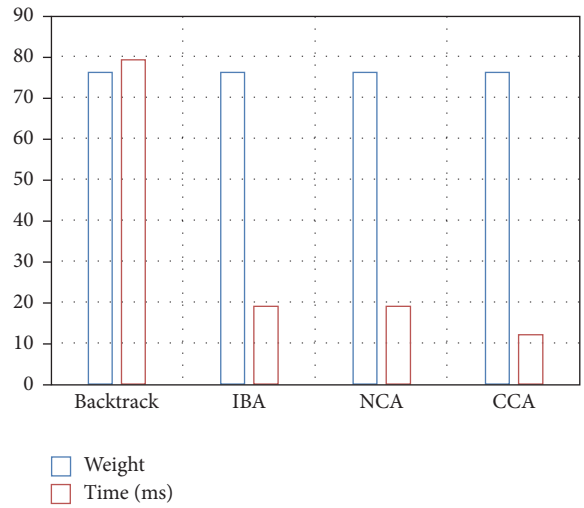


Figure 4: Experimental results of Experiment 2.

of CCA is the highest while IBA and NCA have a similar efficiency because of few remote nodes.

Experiment 3. Total nodes are 30; must-pass nodes are 10; edges are 135.

Figure 5 shows the experimental result from Experiment 3 and it presents the fact that the superiority of CCA proves obvious as graph complexity gradually improves.

Experiment 4. Total nodes are 40; must-pass nodes are 10; edges are 229.

Figure 6 shows the experimental result from Experiment 4 and it presents the fact that backtracking method indicates low efficiency if complexity of the graph is even higher; in contrast, CCA efficiency performs reasonably well.

Experiment results have shown that IBA has a higher efficiency than backtracking method judged by either weights

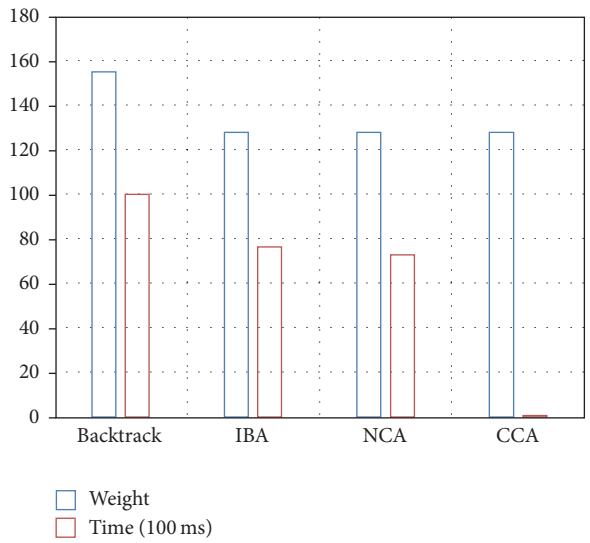


Figure 5: Experimental results of Experiment 3.

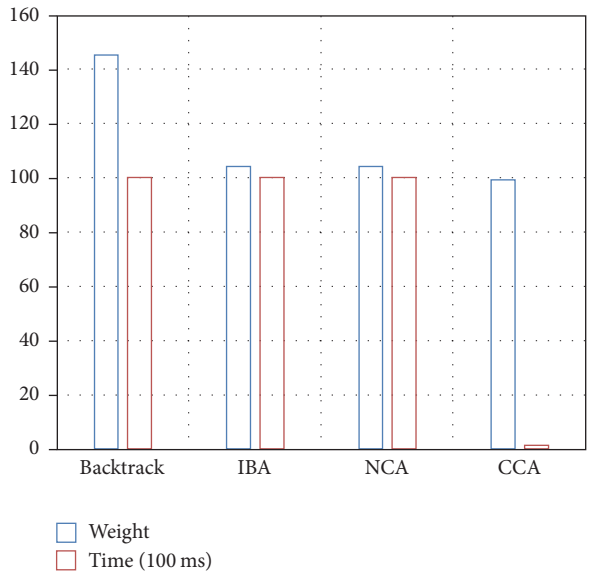


Figure 6: Experimental results of Experiment 4.

or search time. NCA shows only a slight advantage over IBA because remote nodes in the graph are very limited. In particular, judging from all dimensions, CCA has proved significant quality in searching the results with superior efficiency to other algorithms, indicating the effectiveness of CCA in solving such problems.

5.2.2. CCA and ICCA Comparison. It is observed from the previous four experiments that the respective efficiency of backtracking method, IBA, and NCA decreases drastically as the sum of nodes increases. Therefore, there is no research value to add up more nodes to the graph. This section continues to compare between CCA and ICCA.

Experiment environment will remain the same as those of Experiments 1–4; experiment will gradually increase total

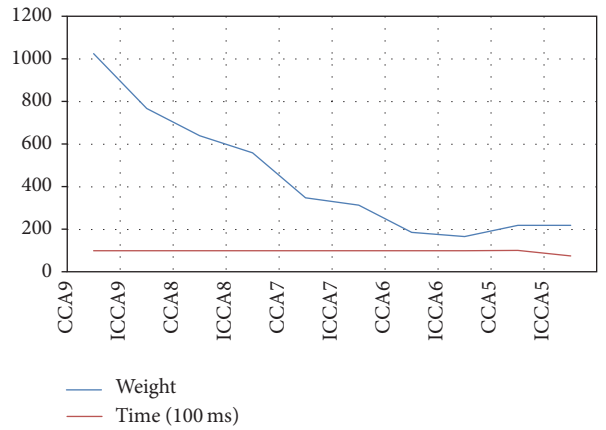


Figure 7: Experimental results of Experiments 5–9.

nodes and edges, while the size of intermediate nodes set will also increase. Comparison will be based on the following five experiments.

Experiment 5. Total nodes are 60, must-pass nodes are 10, and edges are 285.

Experiment 6. Total nodes are 100, must-pass nodes are 15, and edges are 516.

Experiment 7. Total nodes are 200, must-pass nodes are 20, and edges are 997.

Experiment 8. Total nodes are 400, must-pass nodes are 28, and edges are 2178.

Experiment 9. Total nodes are 600, must-pass nodes are 50, and edges are 3418.

Figure 7 shows the experimental results which have indicated that compared to CCA, ICCA obtains better solutions. Therefore, the improved strategy in Section 4.3 is proved to be effective.

6. Conclusion

Problems like postman problem, traveler problem, bus line design, network routing problem, and other similar cases can be abstracted as the path finding graph model as discussed in this study. IBA and NCA are applicable to medium-sized problems. NCA is recommended to solve graphs that contain many remote nodes, while CCA and ICCA are more efficient in dealing with large-scale problems with great algorithm complexity. Additionally, ICCA is able to promote search efficiency when subnodes are readjusted.

As the size of problem becomes larger, CCA and ICCA may not be able to search the whole solution space completely with the optimal solution within a given time. In this case, the compression idea will be integrated into heuristic algorithms such as genetic algorithm and ant colony algorithm to expect a far more efficient search algorithm so as to resolve routing problems with larger scales.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[2] D.-Y. Zhang, W.-L. Wu, and C.-F. Ouyang, "Top-k shortest-path query on RDF graphs," *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, vol. 43, no. 8, pp. 1531–1537, 2015.

[3] H. Y. Cao, Y. Yuan, and Z. Q. Liu, "Routing algorithm for WSNs based on residual energy of node and the maximum angle," *Transducer & Microsystem Technologies*, 2015.

[4] L.-Y. Feng, L.-W. Yuan, W. Luo, R.-C. Li, and Z.-Y. Yu, "Geometric algebra-based algorithm for solving nodes constrained shortest path," *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, vol. 42, no. 5, pp. 846–851, 2014.

[5] W. Zhao, R. Lun, C. Gordon et al., "A human-centered activity tracking system: toward a healthier workplace," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, pp. 343–355, 2017.

[6] T. Li, Y. Liu, L. Gao, and A. Liu, "A cooperative-based model for smart-sensing tasks in fog computing," *IEEE Access*, vol. 5, pp. 21296–21311, 2017.

[7] Y.-H. Qi, Y.-G. Cai, H. Cai, Y.-L. Tang, and W.-X. Lv, "Chaotic Hybrid Discrete Bat AIgorithm for TraveIing SaIesman ProbIem," *Acta Electronica Sinica*, vol. 44, no. 10, pp. 2543–2547, 2016.

[8] Y. Z. Wang, Y. Chen, and J.-S. Zhang, "Novel Fruit Fly Algorithm Based on Learning and Memory for Solving Traveling Salmesman Problem," *Journal of Chinese Computer Systems*, vol. 37, no. 12, pp. 2722–2726, 2016.

[9] C. Yang, X. Zhang, C. Zhong et al., "A spatiotemporal compression based approach for efficient big data processing on Cloud," *Journal of Computer and System Sciences*, vol. 80, no. 8, pp. 1563–1583, 2014.

[10] X. Luo, J. Liu, D. D. Zhang, and X. Chang, "A large-scale web QoS prediction scheme for the Industrial Internet of Things based on a kernel machine learning algorithm," *Computer Networks*, vol. 101, pp. 81–89, 2016.

[11] X. Luo, J. Deng, J. Liu, W. Wang, X. Ban, and J. Wang, "A quantized kernel least mean square scheme with entropy-guided learning for intelligent data analysis," *China Communications*, vol. 14, no. 7, pp. 127–136, 2017.

[12] A. Fernández-Fernández, C. Cervelló-Pastor, and L. Ochoa-Aday, "Improved Energy-Aware Routing Algorithm in Software-Defined Networks," in *Proceedings of the 41st IEEE Conference on Local Computer Networks, LCN 2016*, pp. 196–199, UAE, November 2016.

[13] N. Li, J.-F. Martínez, and V. H. Díaz, "The balanced cross-layer design routing algorithm in wireless sensor networks using fuzzy logic," *Sensors*, vol. 15, no. 8, pp. 19541–19559, 2015.

[14] L. Lei, W. F. Li, and H. J. Wang, "Path optimization of wireless sensor network based on genetic algorithm," *Journal of University of Electronic Science & Technology of China*, vol. 38, no. 2, pp. 227–230, 2009.

[15] X. Luo, Y. Lv, M. Zhou, W. Wang, and W. Zhao, "A laguerre neural network-based ADP learning scheme with its application to tracking control in the Internet of Things," *Personal and Ubiquitous Computing*, vol. 20, no. 3, pp. 361–372, 2016.

[16] X. Luo and X. Chang, "A novel data fusion scheme using grey model and extreme learning machine in wireless sensor networks," *International Journal of Control, Automation, and Systems*, vol. 13, no. 5, 2015.

Hindawi
Scientific Programming
Volume 2017, Article ID 3704525, 8 pages
https://doi.org/10.1155/2017/3704525

Research Article

Adaptive Ensemble Method Based on Spatial Characteristics for Classifying Imbalanced Data

Lei Wang,[1,2] Lei Zhao,[1] Guan Gui,[1] Baoyu Zheng,[1] and Ruochen Huang[1]

[1]*National Local Joint Engineering Research Center for Communication and Network Technology, Nanjing University of Posts and Telecommunications, Nanjing, China*
[2]*The State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China*

Correspondence should be addressed to Guan Gui; guiguan@njupt.edu.cn

Received 15 September 2017; Accepted 3 December 2017; Published 26 December 2017

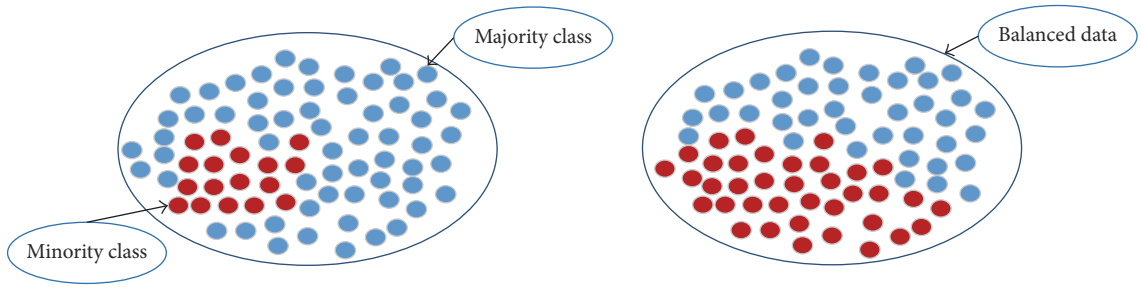
Academic Editor: Anfeng Liu

The class imbalance problems often reduce the classification performance of the majority of standard classifiers. Many methods have been developed to solve these problems, such as cost-sensitive learning methods, synthetic minority oversampling technique (SMOTE), and random oversampling (ROS). However, the existing methods still have some problems due to the possible performance loss of useful information and overfitting. To solve the problems, we propose an adaptive ensemble method by using the most advanced feature of self-adaption by considering an average Euclidean distance between test data and training data, where the average distance is calculated by k-nearest neighbors (KNN) algorithm. Simulation results are provided to confirm that the proposed method has a better performance than existing ensemble methods.

1. Introduction

Imbalanced data refers to a data set that has great differences in the number of the classes. Currently, imbalanced data has been applied to real-word domain and plays a key role in civilian and government applications, such as text classification [1], facial age estimation [2], speech recognition [3], and governmental decision-making support systems [4]. The research of imbalanced data is of great significance in the fields of credit fraud, data mining, and illegal account invasion. Hence, more and more researchers pay great attentions to class imbalanced issues due to the fact that the traditional classification of imbalanced data processing is not suitable for classifying minority classes. Imbalanced problem also caught the attention of related areas such as machine learning and data mining [4].

The present study focuses more on binary class imbalance problem, where data set is sorted into majority classes and minority classes. In the data set, the traditional balanced data means that the numbers of each class are equal, and the imbalanced data means that the numbers of the various classes are significantly different. The details of binary class imbalanced and balanced data are shown in Figures 1(a) and 1(b). The traditional classification algorithms, such as naive Bayes [5], random forest [6], K-nearest neighbors (KNN) [7], and RIPPER [8], aim at generating models that can optimize the accuracy over classification, but they neglect the minority class. In order to solve the problem mentioned above, many methods have been proposed about binary class imbalanced data in data level and algorithm level, respectively. In the data level, the major idea is to transform imbalanced into balanced data mainly by using sampling method or to create new examples for imbalanced into balanced data, such as SMOTE and ROS, while the algorithm-level solutions primarily include ensemble learning methods [9] and cost-sensitive analysis. Generally speaking, these methods solved the problem of imbalanced data in the accuracy of minority classes. However, there still exist some drawbacks in these traditional imbalanced data classification methods for handling binary class imbalanced data problems [10]. For example, boosting and bagging based ensemble methods may lose some valuable information in the iteration process owing to the use of sampling methods. As a result, this may cause the data overfitting problem. Moreover, it is hard to



(a) An example of binary imbalanced data, where majority class is 58 and minority class is 17

(b) An example of binary balanced data, where majority data is equal to minority data

Figure 1: A figure illustration of imbalanced data (a) and balanced data (b).

get the optimized misclassification cost in the cost-sensitive learning methods, and different misclassification costs lead to different classification results. Hence, the classification results are not stable.

To overcome the above problems, we proposed an adaptive ensemble method that is an improvement of existing ensemble method [10]. *Our main idea is to transform imbalanced binary problem into multiple balanced problems, which neither reduce the number of majority classes nor increase that of the minority classes*. Then we build multiple base classifiers to deal with these balanced problems and lastly we use an adaptive ensemble rule to assemble the base classification results obtained from base classifiers. Common ensemble rules including Max Rule, Majority Vote Rule, Product Rule, Min Rule, and Sum Rule were proposed in [11] and several novel ensemble rules including MaxDistance Rule, MinDistance Rule, ProDistance Rule, MajDistance Rule, and SumDistance Rule were put forward in [10]. In [10], the test results indicate that their methods have a better performance compared with many conventional imbalanced data processing methods over some standard imbalanced data sets. Meanwhile, the results of their experiments proved that SplitBal + MaxDistance has a better performance than other combinations. Throughout this paper, SplitBal + MaxDistance is referred to as SMD.

We have two improvement points for SMD and we define it as SplitBal + MaxDistanceandAvePr (SMDA), which shares the same process with SMD except the ensemble rule. By using base classification algorithms including naive Bayes, random forest, logistic regression, and SVC [12], empirically, our proposed method is evaluated over 38 highly imbalanced data sets. After that, the numerical results show that our method is superior to SMD.

The rest of this paper is organized as follows. Section 2 introduces the works related to our research. Section 3 shows the proposed method. Section 4 reports our experimental procedure, describes details on the setup of experiments, and analyzes the processed data results. Finally, in Section 5, we summarize the study and draw the conclusion.

2. Related Work

Over the past decades, the imbalanced data problem has always been a difficult problem in data mining. There are also other data characteristics such as data shift [13] and class overlapping [14], which can influence the performance of conventional classification algorithms for dealing with imbalanced problems. However, we still focus on the imbalance characteristic between classes.

Table 1: Related work list.

Data level	[9, 13, 14]
Algorithm level	
Ensemble learning	[10, 11]
Cost-sensitive learning	[15]

So far, many measures have been proposed to solve the binary class imbalance problem [10–14, 16–21]. These measures can be broadly by data level and algorithm level, as shown in Table 1. The existing measures can adapt the imbalanced class in algorithm level, while preprocess of measures can adjust data from being imbalanced to balanced in data level. Our methods could be regarded as in algorithm level; in this section, we will introduce some methods that belong to algorithm level.

The algorithm level includes cost-sensitive learning, ensemble learning, and recognition-based learning. (1) Cost-sensitive learning approaches obtain the lowest classification error by adjusting the class misclassification cost. MetaCost [15] is a kind of this algorithm, which uses cost-sensitive procedure to make the classification algorithm cost-sensitive. (2) Ensemble learning is used to reduce the variance and bias by integrating the results of many classification algorithms on imbalanced data. Representatively, boosting can adaptively identify the samples, which is classified as error, so it can obtain a good performance on imbalance problem. Bagging improves the classification performance by processing the base classifiers. (3) Autoassociation, RIPPER, and recognition based learning provide the discrimination model created on the examples of the target class alone which have been certified to be effective in dealing with high-dimensional and complicated binary imbalanced data.

However, these ensemble methods may have some unavoidable drawbacks such as changing the raw data spatial distribution or lead to overfitting caused by sampling methods. In addition, these ensemble algorithms may lose the connection between the test data and training data. In other

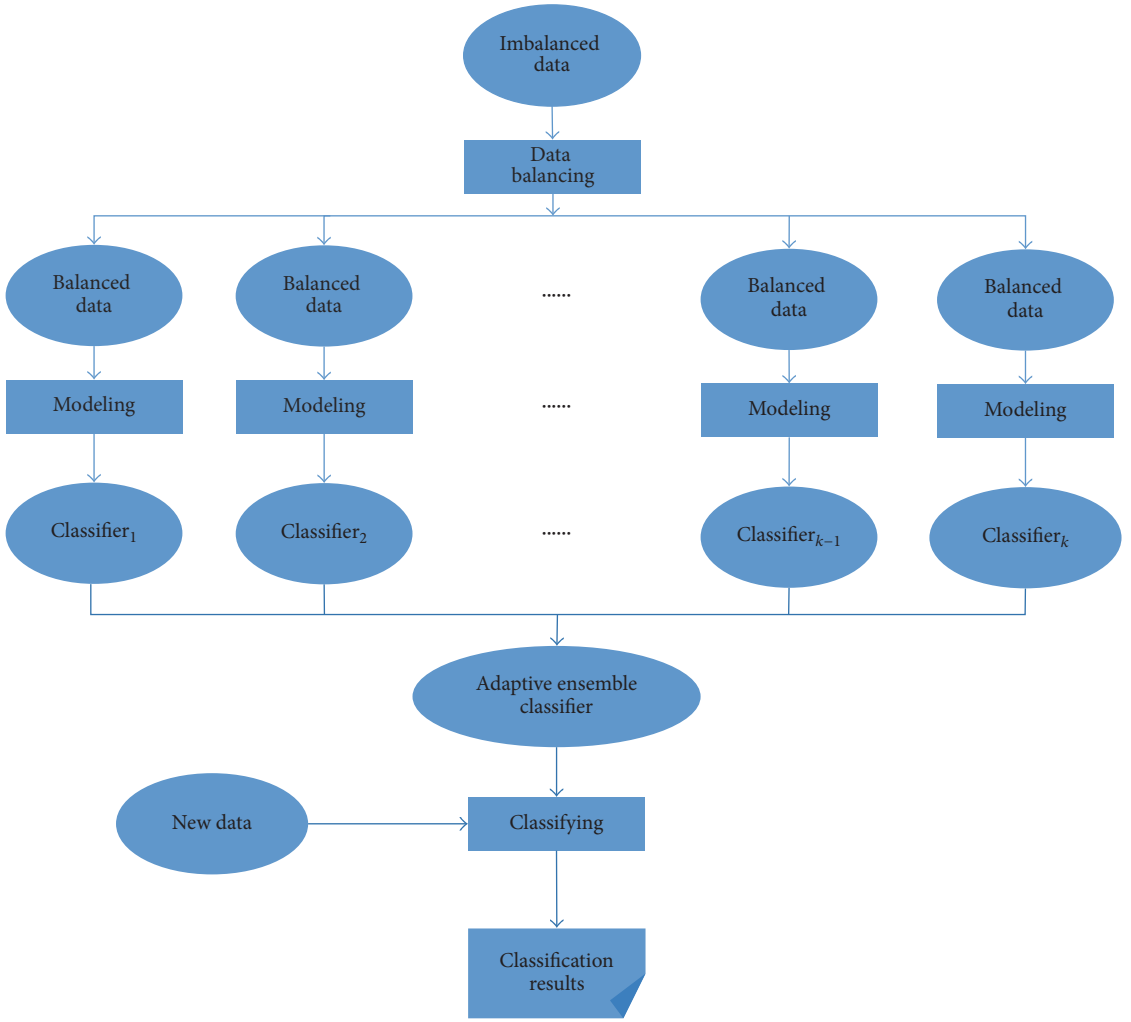


Figure 2: Framework of our proposed adaptive ensemble method for handling binary imbalanced data.

words, test data can be classified as the class which is closer than the spatial distribution distance. Our method avoids some weaknesses of these traditional imbalanced problems processing methods mentioned previously by transforming the imbalanced problem into several balanced ones; thus it is not like the existing imbalance problems handling methods. Furthermore, our method takes into account the distance between data factors in the ensemble rule, because, in theory, the closer they are, the more similar they are, and our ensemble method is adaptive, which is different from SMD.

3. Our Proposed Method

Our proposed method includes three parts: data balancing, modeling, and classifying. Figure 2 describes the frame of the proposed method. For data balancing, in the process of our method, we first divide the majority data set into several parts which are equal to the amount of minority class data. Then we combine the part with the minority class into a new balanced data set. So many balanced data sets are received. For modeling, next, each new balanced data set is used to create a base classifier with a given base classification algorithm. As for classifying, lastly, these base classification results are put into an adaptive ensemble classifier to classify test data. In the modeling component, we directly apply a base algorithm to every balanced data set. Subsequently, we will introduce two procedures of *data balancing* and *classifying* as follows.

3.1. Data Balancing. Existing measures to balance the imbalanced data usually lead to the loss of information as well as overfitting. Therefore, it can be realized to transform imbalanced data set into multiple balanced data sets without importing noise data or lessening the raw data. It is well known that the majority of class data sets are usually more than the minority in an imbalanced data set. So we divide the majority class data set into multiple sets, and each set is equal to minority class in number. Considering the similarities of a class, we can split the majority class data set into multiple sets (SplitBal). Then each set is added to the minority class data

Table 2: The strategies and descriptions for MaxDistance Rule.

Rule	Strategy
MaxDistance	$R_1 = \arg\max_{1\le i\le k} \frac{p_{i1}}{D_{i1}+1}$, $R_2 = \arg\max_{1\le i\le k} \frac{p_{i2}}{D_{i2}+1}$

Table 3: The strategies for AdaptiveMaxDistance Rule.

Rule	Strategy
AdaptiveMaxDistance	$R_1 = \arg\max_{1\le i\le k} \frac{p_{i1}}{D_{i1}+\alpha} + \text{average}(P_{i1})$ $R_2 = \arg\max_{1\le i\le k} \frac{p_{i2}}{D_{i2}+\alpha} + \text{average}(P_{i2})$

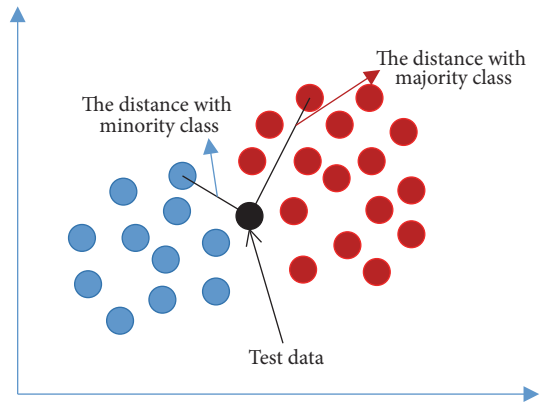


Figure 3: Figure illustration of the average distance.

set to build a new balanced data set. Lastly, we could obtain multiple balanced data sets.

3.2. Classifying. After modeling, we can build multiple classifiers with the processed balanced data acquired from data balancing. Then we can get some classification results from these classifiers. Next we combine these classification results together. Like [9], we make some assumptions as follows: assume that there are k binary class data sets and two class labels; the class labels are C_1 and C_2. Then we could get K base classifiers with a given base algorithm. For the ith classifier ($1 \le i \le k$), it will classify the test data as C_1 with the probability P_{i1} and as C_2 with the probability P_{i2}. Moreover, R_1 and R_2 represent the ultimate ensemble results for classes C_1 and C_2, respectively. Five ensemble rules and their description and details are shown in [10]. But these ensemble rules just adopt the results of the classification while overlooking the connection between the test data and training data. The test data tends to be grouped into the class whose average Euclidean instances are closer to the test data. A new multilabel classifier that uses neighbor distance was mentioned in [22]. Considering the average Euclidean instances between the test data and the training data, five novel ensemble rules were proposed in [9]. In these ensemble methods, D_{ij} ($1 \le i \le k$, $1 \le j \le 2$) indicate the average Euclidean distance between new data and the data with class label C_j in the ith data. From Figure 3, we can learn the process of obtaining D_{ij} with KNN. The details of MaxDistance are shown in Table 2.

However, in the rules $R_1 = \arg\max_{1\le i\le k}(p_{i1}/(D_{i1}+1))$ and $R_2 = \arg\max_{1\le i\le k}(p_{i2}/(D_{i2}+1))$, a fixed value should be added in the denominator with the purpose of preventing the average Euclidean distance from being equal to 0. Then the results of experiment in [10] show that the value can be defined anywhere from 0 to 1, so they add the distance with 1. But we argue that the value added to average in the denominator could be defined with 0, and the value also should be adaptive from 0 to 1 with different classification algorithms. The reasons are as follows. (1) The average distance D_{ij} could not be 0, because from Figure 3 we can know that it is impossible for the new data to be the same as all the train data. (2) When we use different base classification algorithms, the value added to average in the denominator should be mutative. So we define the added value as α, which ranges from 0 to 1. (3) From the MaxDistance Rule, we can find that the effect of P_{il} is weak even though it has been considered as the best important decision element in most traditional algorithms. And in [22], we find that EMLA (average of P_{il}) always has a better performance than other ensemble rules, so we combine the EMLA with $R_1 = \arg\max_{1\le i\le k}(p_{i1}/(D_{i1}+\alpha))$ and $R_2 = \arg\max_{1\le i\le k}(p_{i2}/(D_{i2}+\alpha))$ as shown in Table 3. Finally, the classification results R_1 and R_2 are obtained with the ensemble rules in Tables 2 and 3, respectively. If $R_1 \ge R_2$, the test data is considered as C_1; otherwise it is considered as C_2.

4. Numerical Simulation

In this paper, we have adopted 38 public imbalanced data sets which came from Keel data set repository [23]. The details of these 38 data sets are shown in Table 4, including imbalance radio, total attributes (ATT), total number of data sets, and the number of minority (positive) class data sets. For more detailed information about the employed data sets, interested authors are referred to http://sci2s.ugr.es/keel/imbalanced.php.

We use the 5-fold cross-validation strategy in the following experiment. Four different base classification algorithms, naive Bayes, random forest, logistic regression, and SVC, were selected as base classifiers. We use AUC [24] as our algorithm metric which has more advantages than G-Mean and F-Measure [25]. In our experiment, every AUC of every data set will be tested repeatedly and then take an average.

Our study in this paper made up two experiments. The first experiment is to determine the added value in our rule. Then the second experiment is to compare the proposed method SMDA with SMD method when handling the imbalanced binary problems using different base classification algorithms.

Experiment 1. We first use the data set yeast3 (shown in Table 5) to test the AUC value of our method by using different values in our ensemble rules. Then we choose the

Table 4: Statistic summary of the 38 highly imbalanced data sets.

Data set	ATT	Instance	Minority	IR
(1) Yeast3	9	1484	163	8.10
(2) Ecoli03	8	336	35	8.60
(3) Yeast2vs4	9	514	51	9.08
(4) Ecoli067vs35	8	222	22	9.09
(5) Ecoli0234vs5	8	202	20	9.10
(6) Glass015vs2	10	172	17	9.12
(7) Yeast0359vs78	9	506	50	9.12
(8) Yeast0256vs3789	9	1004	99	9.14
(9) Yeast02579vs368	9	1004	99	9.14
(10) Ecoli046vs5	7	203	20	9.15
(11) Yeast1289vs7	9	947	30	30.57
(12) Ecoli0267vs35	8	224	22	9.18
(13) Glass04vs5	10	92	9	9.22
(14) Ecoli0346vs5	8	205	20	9.25
(15) Ecoli0347vs56	8	257	25	9.28
(16) Yeast05679vs4	9	528	51	9.35
(17) Vowel0	14	988	90	9.98
(18) Ecoli067vs5	7	220	20	10.00
(19) Glass016vs2	10	192	17	10.29
(20) Led7digit	8	443	37	10.97
(21) Ecoli01vs5	7	240	20	11.00
(22) Glass06vs5	10	108	9	11.00
(23) Glass0146vs2	10	205	17	11.06
(24) Glass2	10	214	17	11.59
(25) Ecoli0147vs56	7	332	25	12.28
(26) Ecoli0146vs5	7	280	20	13.00
(27) Shuttlec0vs4	10	1892	123	13.87
(28) Yeast1vs7	8	459	30	14.30
(29) Glass4	10	214	13	15.46
(30) Ecoli04	8	336	20	15.80
(31) Pageblock13vs4	11	472	28	15.86
(32) Glass016vs5	10	184	9	19.44
(33) Glass5	10	214	9	22.78
(34) Yeast2vs8	9	482	20	23.10
(35) Yeast4	9	1484	51	28.10
(36) Yeast5	9	1484	44	32.73
(37) Ecoli0137vs26	8	281	7	39.14
(38) Yeast6	9	1484	35	41.40

Table 5: The data set of Yeast3.

Data set	ATT	Instances	Minority	IR
Yeast3	9	1484	163	8.10

fixed value which can make the best AUC. In experiment, the values are 0, 0.2, 0.4, 0.6, 0.8, and 1.0. From Figure 4, we can know that α should be defined as 1, 0, 1, and 0 when the base classification algorithms are naive Bayes, random forest, logistic regression, and SVC, respectively.

Experiment 2. Performance results are evaluated in comparisons of SDMA and SMD. For every imbalanced data set, the detailed AUC values for both methods with the four different base classification algorithms are shown in Table 6. The end of the row represents the average of AUC values of the two methods with each classification algorithm. From Figure 5, we can observe that there are 33 AUC values of SMDA which are greater than or equal to SMD by using logistic regression. In addition, 25 AUC values of SMDA are greater than SDMA using SVC, while they are greater than or equal to SMD using random forest. It is noticed that 30 AUC values of SMDA are great than or equal to SMD using naive Bayes. In Figure 6, we

Table 6: AUC value for SMDA and SMD using different classification algorithms.

DATA	Logistic regression		SVC		Random forest		Naive Bayes	
	SMDA	SMD	SMDA	SMD	SMDA	SMD	SMDA	SMD
Yeast3	0.9617	0.9600	0.9521	0.8529	0.9757	0.9773	0.9093	0.8355
Ecoli03	0.9259	0.9309	0.9211	0.8997	0.9415	0.9349	0.8955	0.9037
Yeast2vs4	0.9206	0.9192	0.9211	0.8655	0.9755	0.9720	0.8798	0.8487
Ecoli067vs35	0.9497	0.8697	0.8508	0.8677	0.9551	0.9594	0.8433	0.8326
Ecoli0234vs5	0.9398	0.8809	0.9396	0.9425	0.9873	0.9845	0.7885	0.8595
Glass015vs2	0.5556	0.5644	0.7600	0.7033	0.8033	0.7678	0.7022	0.6877
Yeast0359vs78	0.8044	0.7877	0.7012	0.5987	0.8334	0.8457	0.7312	0.6032
Yeast0256vs3789	0.8443	0.8429	0.8190	0.7928	0.8473	0.8393	0.7156	0.6893
Yeast02579vs368	0.9380	0.9379	0.9142	0.8958	0.9579	0.9589	0.9214	0.8251
Ecoli046vs5	0.9593	0.9343	0.9283	0.9523	0.9903	0.9861	0.8924	0.8792
Ecoli0267vs35	0.9198	0.8932	0.8315	0.8797	0.9373	0.9382	0.8369	0.8286
Glass04vs5	0.9538	0.9470	0.9667	0.9667	1.0000	1.0000	0.9875	0.9875
Ecoli0346vs5	0.9444	0.8375	0.9278	0.9486	0.9833	0.9819	0.8931	0.8986
Ecoli0347vs56	0.9388	0.8982	0.9178	0.9371	0.9720	0.9712	0.8896	0.8957
Yeast05679vs4	0.8479	0.8437	0.8333	0.7439	0.9142	0.9134	0.7835	0.7151
Vowel0	0.9833	0.9826	0.9992	0.9997	0.9991	0.9988	0.9715	0.9340
Ecoli067vs5	0.8910	0.8359	0.9308	0.9231	0.9654	0.9731	0.8064	0.8385
Glass016vs2	0.6529	0.6265	0.7211	0.6760	0.7824	0.7770	0.6897	0.6843
Led7digit	0.9530	0.9525	0.9619	0.9615	0.9549	0.9538	0.9462	0.9427
Ecoli01vs5	0.9767	0.9580	0.9488	0.9453	0.9872	0.9918	0.8581	0.8186
Glass06vs5	0.9094	0.8404	0.9839	0.9739	1.0000	1.0000	1.0000	1.0000
Glass0146vs2	0.7048	0.6663	0.7336	0.7324	0.8276	0.8480	0.7287	0.7169
Glass2	0.6923	0.6752	0.8447	0.8175	0.8213	0.8011	0.7252	0.7293
Ecoli0147vs56	0.9393	0.9016	0.9236	0.9255	0.9736	0.9742	0.8076	0.8045
Ecoli0146vs5	0.9412	0.9284	0.9480	0.9500	0.9863	0.9882	0.8076	0.8045
Shuttlec0vs4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9757	0.9757
Yeast1vs7	0.8273	0.8268	0.7849	0.6529	0.8306	0.8368	0.7835	0.6054
Glass4	0.9316	0.8585	0.9595	0.9662	0.9663	0.9655	0.8688	0.8814
Ecoli04	0.9831	0.9815	0.9911	0.9100	0.9928	0.9928	0.9203	0.9428
Pageblock13vs4	1.0000	0.9985	0.7889	0.9119	0.9984	0.9968	0.9623	0.8959
Glass016vs5	0.8912	0.8765	0.9526	0.9559	0.9912	0.9882	1.0000	1.0000
Glass5	0.8725	0.8600	0.9350	0.9230	0.9925	0.9875	0.9950	0.9950
Yeast2vs8	0.8574	0.8546	0.7663	0.7256	0.8806	0.8795	0.8038	0.7758
Yeast4	0.8740	0.8743	0.8692	0.8154	0.9307	0.9277	0.8368	0.7837
Yeast1289vs7	0.7782	0.7776	0.6606	0.5877	0.7960	0.7974	0.7741	0.5185
Yeast5	0.9803	0.9859	0.9827	0.9373	0.9912	0.9912	0.9861	0.9524
Ecoli0137vs26	0.9164	0.9385	0.9443	0.9702	0.9611	0.9684	0.9572	0.9084
Yeast6	0.9254	0.9235	0.9273	0.8529	0.9482	0.9478	0.9516	0.9394
Average	0.8919	0.8729	0.8879	0.8674	0.9382	0.9370	0.8654	0.8386

can see that the average AUC values of our method are greater than SMD overall. Therefore, we can obtain that that SMDA has a better performance than SMD in dealing with the data sets mentioned above.

5. Conclusion

An adaptive ensemble method based on spatial characteristics for dealing with the binary class imbalanced problems has been given in this paper. Different from the existing methods mentioned in this paper, our method firstly uses an adaptive ensemble rule for dealing with imbalanced binary problem. Furthermore, our method neither alters the raw data distribution nor suffers from unexpected mistakes or data loss.

Our method applies random splitting to the majority class instances to transform the imbalanced binary class data into multiple balanced binary class data. After that, we use a base classification algorithm to build multiple base

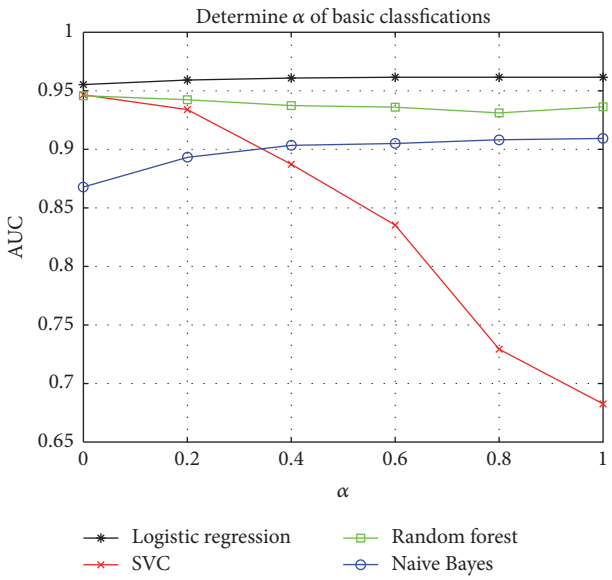


Figure 4: The AUC of different algorithms.

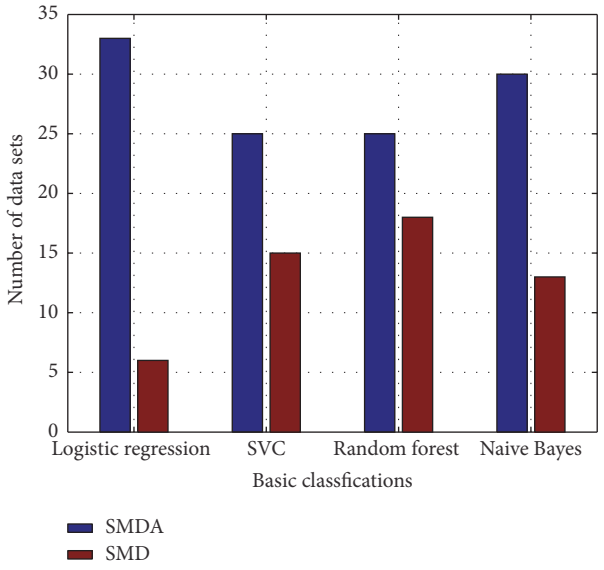


Figure 5: The performance comparison of SMDA (our proposed) and SMD (existing method [10]).

classifiers. Finally, we use the proposed adaptive ensemble rule to assemble the classification results received from base classifiers. The experimental results show that (i) the added variable value to the distance in our methods is adaptive, which changes with the classification algorithm and ranges from 0 to 1, and (ii) our ensemble rule SMDA has a better performance than SMD, so we could obtain that the proposed method currently performs better than the existing methods mentioned in this paper.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

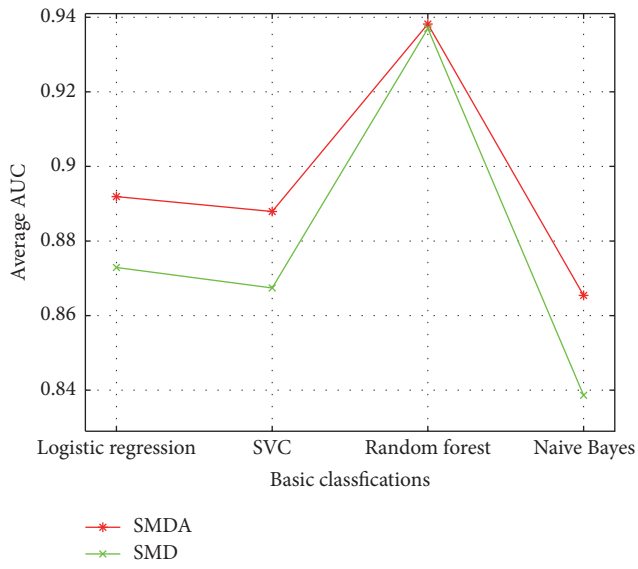


Figure 6: The average value of AUC.

Authors' Contributions

Lei Wang and Lei Zhao conceived and designed the experiments; Lei Zhao and Guan Gui performed the experiments; Baoyu Zheng and Ruochen Huang analyzed the data and also gave comprehensive comments and suggestions; Lei Wang wrote the paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants nos. 61671253 and 61401069), the Priority Academic Development Program of Jiangsu Higher Education Institutions, China, the Major Projects of the Natural Science Foundation of the Jiangsu Higher Education Institutions (16KJA510004), the Open Research Fund of the State Key Laboratory of Integrated Services Networks, Xidian University (ISN17-04), and the Open Research Fund of National Local Joint Engineering Research Center for Communication and Network Technology, Nanjing University of Posts and Telecommunications (TXKY17005).

References

[1] M. D. D. Castillo and J. I. Serrano, "A multi strategy approach for digital text categorization," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 15–32, 2004.

[2] W.-L. Chao, J.-Z. Liu, and J.-J. Ding, "Facial age estimation based on label-sensitive learning and age-oriented regression," *Pattern Recognition*, vol. 46, no. 3, pp. 628–641, 2013.

[3] A. An, N. Cercone, and X. Huang, "A case study for learning from imbalanced data sets," in *Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence*, vol. 2056 of *Lecture Notes in Computer Science*, pp. 1–15, Springer, Berlin, Germany, 2001.

[4] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[5] Z. Zheng, "Naive bayesian classifier committees," *Lecture Notes in Computer Science*, vol. 1398, pp. 196–207, 1998.

[6] A. Liaw and M. Wiener, "Classification and regression by random forest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[7] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, article 1883, 2009.

[8] W. W. Cohen, "Fast effective rule induction," in *12th International Conference on Machine Learning*, pp. 115–123, Miami, Fl, USA, 2013.

[9] Z. Sun, Q. Song, and X. Zhu, "Using coding-based ensemble learning to improve software defect prediction," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 6, pp. 1806–1817, 2012.

[10] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, and Y. Zhou, "A novel ensemble method for classifying imbalanced data," *Pattern Recognition*, vol. 48, no. 5, pp. 1623–1637, 2015.

[11] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[12] C. Ullrich, "Support vector classification," in *Forecasting and Hedging in the Foreign Exchange Markets*, pp. 65–82, Springer, Berlin, Germany, 2009.

[13] V. López, A. Fernández, and F. Herrera, "On the importance of the validation technique for classification with imbalanced datasets: addressing covariate shift when data is skewed," *Information Sciences*, vol. 257, no. 2, pp. 1–13, 2014.

[14] R. Alejo, R. M. Valdovinos, V. García, and J. H. Pacheco-Sanchez, "A hybrid method to face class overlap and class imbalance on neural networks and multi-class scenarios," *Pattern Recognition Letters*, vol. 34, no. 4, pp. 380–388, 2013.

[15] P. Domingos, "Metacost:a general method for making classifiers cost sensitive," in *Proceedings of the International Conference on Knowledge Discovery & Data Mining*, pp. 155–164, San Diego, Cal, USA, 1999.

[16] A. Liu, Z. Chen, and N. N. Xiong, "An adaptive virtual relaying set scheme for loss-and-delay sensitive WSNs," *Information Sciences*, vol. 424, pp. 118–136, 2018.

[17] M. Zhou, M. Zhao, A. Liu, M. Ma, T. Wang, and C. Huang, "Fast and efficient data forwarding scheme for tracking mobile targets in sensor networks," *Symmetry*, vol. 9, no. 11, article 269, 2017.

[18] D. Wu, J. Wang, R. Q. Hu, Y. Cai, and L. Zhou, "Energy-efficient resource sharing for mobile device-to-device multimedia communications," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, pp. 2093–2103, 2014.

[19] D. Wu, L. Zhou, Y. Cai, R. Hu, and Y. Qian, "The role of mobility for D2D communications in LTE-advanced networks: energy vs. bandwidth efficiency," *IEEE Wireless Communications Magazine*, vol. 21, no. 2, pp. 66–71, 2014.

[20] L. Zhou, R. Q. Hu, Y. Qian, and H.-H. Chen, "Energy-spectrum efficiency tradeoff for video streaming over mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 5, pp. 981–991, 2013.

[21] L. Zhou, "Mobile device-to-device video distribution: theory and application," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 12, no. 3, article 38, pp. 1253–1271, 2016.

[22] M. A. Tahir, J. Kittler, and A. Bouridane, "Multilabel classification using heterogeneous ensemble of multi-label classifiers," *Pattern Recognition Letters*, vol. 33, no. 5, pp. 513–523, 2012.

[23] J. Alcal, A. Fernndez, J. Luengo, J. Derrac, S. Garcła, and L. Snchez, "KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic & Soft Computing*, vol. 17, no. 2, pp. 255–287, 2011.

[24] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 3, pp. 299–310, 2005.

[25] H. He and Y. Ma, *Imbalanced Learning: Foundations, Algorithms, and Applications*, John Wiley and Sons, Hoboken, NJ, USA, 2013.

Hindawi
Scientific Programming
Volume 2017, Article ID 4379141, 9 pages
https://doi.org/10.1155/2017/4379141

Research Article

A Novel Hybrid Similarity Calculation Model

Xiaoping Fan,[1,2] Zhijie Chen,[1] Liangkun Zhu,[3] Zhifang Liao,[3] and Bencai Fu[3]

[1]*School of Information Science and Engineering, Central South University, Hunan, China*
[2]*Hunan University of Finance and Economics, Hunan, China*
[3]*School of Software, Central South University, Hunan, China*

Correspondence should be addressed to Zhifang Liao; zfliao@csu.edu.cn

Received 25 August 2017; Accepted 12 November 2017; Published 4 December 2017

Academic Editor: Longxiang Gao

This paper addresses the problems of similarity calculation in the traditional recommendation algorithms of nearest neighbor collaborative filtering, especially the failure in describing dynamic user preference. Proceeding from the perspective of solving the problem of user interest drift, a new hybrid similarity calculation model is proposed in this paper. This model consists of two parts, on the one hand the model uses the function fitting to describe users' rating behaviors and their rating preferences, and on the other hand it employs the Random Forest algorithm to take user attribute features into account. Furthermore, the paper combines the two parts to build a new hybrid similarity calculation model for user recommendation. Experimental results show that, for data sets of different size, the model's prediction precision is higher than the traditional recommendation algorithms.

1. Introduction

Traditional collaborative filtering (CF) algorithms usually calculate similarity between users or items based on user-item rating matrix, and in the light of the calculated similarity they choose the nearest neighbor and construct prediction scores to generate recommendation lists. Therefore, the similarity calculation decides the precision and quality of recommendations produced by the heuristic CF algorithm. However, the present traditional heuristic CF recommendation algorithms suffer from a range of problems in similarity calculation, such as the failure in finding changes of user interest; that is, by directly computing similarity on the basis of statistics, it considers user ratings and center ratings only while ignoring other factors when rating, such as user attributes, time weight, and user rating habits.

In order to solve the problems of similarity calculation in traditional heuristic CF recommendation and improve its performance, Luo et al. [1], Anand and Bharadwaj [2], and Lopes et al. [3] proposed the global similarity measure. Based on the traditional similarity algorithms, the global similarity measure takes the transitive relationships among users into account to calculate the global similarity and build the user's nearest neighborhood set. Results of Lopes' experiments indicated that, in case of the extremely sparse data set, the combination of traditional similarity algorithms and the global similarity measure can improve the accuracy of recommendation. Li et al. [4] proposed the concept of fluctuation factor. He considered the influence of fluctuation factors and removed the influence of them by z-score method when computing the similarity between users. Shen et al. [5] proposed a two-stage similarity learning algorithm, in which at the first stage it utilizes PCC to calculate the similarity and obtains the nearest neighbor, and at the second stage it uses the reduced gradient method to learn the similarity, which improves the recommendation accuracy. Gao and Huang [6] proposed the idea based on the model of item gravity attribute. Its similarity calculation contains two parts: one of which is the similarity obtained by the traditional calculation; the other part firstly defines the weight value of the item attribute, and then the initial similarity is calculated by the model of item gravity attribute, and, after the two similarities are weighted, the effect of the rating time is taken into account to calculate the final similarity value.

Starting from different perspectives, the studies above aimed at strengthening the association between users and items to improve the similarity between users or items and get the optimal nearest neighbor set, finally improving

the recommendation accuracy and quality on this basis. However, when strengthening the association between users and items, we can take some factors into account, such as the demographic characteristics of users and the time decay caused by the time-effect of ratings, which have certain effects on the association. It is very effective to consider user attribute features when dealing with the problem of user's cold start.

Therefore, the paper proposes a new similarity calculation method: RIT-UA algorithm. The RIT-UA algorithm consists of two parts: one is the similarities of user rating-interest, which considers the similarities of user rating and interest as well as the changes and effects of the two under the constraints of rating time and confidence coefficient between users; the other part is the similarities of the user attributes, which takes into account the influence of the user attribute feature on the recommendation and calculates the similarity of the user attributes after getting the weight of each attribute feature. In the end, RIT-UA algorithm fits the two parts linearly. The experimental results show that, compared with the traditional methods, the algorithm proposed in this paper can obtain better prediction accuracy.

2. Related Work

In studies of recommendation system, though in recent years the recommender systems have been studied frequently and developed sufficiently, there are still some common problems, such as data sparsity, cold start, and user interest drift. In order to deal with these problems and improve the recommendation precision and accuracy, researchers may take many aspects into account, including the basic user attribute feature and the time and place where the user behavior occurred, and researches about these came into being correspondingly.

Demographic Recommender System (DRS) is an important part of recommender systems. Demographic characteristics can be used to identify the user's type and their preferences, and the system can sort users according to their attribute features and generate recommendations based on the sorting results. DRS plays a great supporting role in dealing with the problems of user cold start and data sparsity. Many of the present studies have proved that user attribute features can improve the accuracy in recommendations. Luo et al. [7] used improved quantized kernel least mean square (EQ-KLMS) algorithm, which improved the efficiency of machine learning and improved the accuracy of weather forecast. Beel et al. [8] elaborate the role of user attribute features in the recommended process and analyze and demonstrate that the user's attribute characteristics have a significant impact on click-through rates on recommender systems. From the perspective of tourism recommendation, Wang et al.'s [9] experiments proved that the combination of machine learning method (Naive Bayes, Bayesian network, and SVM) and demographic characteristics can improve the prediction accuracy of tourism recommendations. Zhao et al. [10] used visual tracking sensors to acquire biometric information and then used machine learning based biometrics to improve the accuracy of recognition. Combined with user attribute features, Al-Shamri [11] constructed five similarity measures, respectively, based on user preference modeling method, and the experimental results showed that the combination of user attribute features improves the recommendation accuracy of recommender systems. Santos et al. [12] applied user attribute features in real recommendation environment to mine and analyze the context constraints in the scene. Chen and He [13] constructed the user demographic vector by the user information, and, on this basis, took the corated item and the item's frequency into account to figure out a new similarity. The experimental results showed that this approach can solve the problem of cold start effectively and improve the recommendation accuracy. Luo et al. [14] achieve QoS prediction with automatic parameter tuning capability by using approximate dynamic programming, through online learning and optimization, without the need for preknowledge or prediction model identification. Then, through the use of a kernel least mean square algorithm [15], the lack of Web services QoS is forecasting. The experimental results show that the method can effectively solve the cold start problem and improve the prediction accuracy.

With the intensive development of recommender systems research, in order to obtain better recommendations and improve recommendation quality, many researchers began to incorporate contextual information into the research of the recommender systems. Relatively speaking, the time information is easier to collect among contextual information, and it provides significant value for researches on improving the diversity of timing sequence of the recommender systems, which has become a hot topic in the current studies [16]. Koren [17] used matrix factorization (SVD), which regards time as an important feature and add it to the feature data set of user-item, and solved the problem of user interest drift effectively. Karatzoglou et al. [18] and Xiong et al.[19] regarded the time information as the third eigenvector, employing the approach of tensor factorization to show the dynamic changes of time. According to the user's rating history, Rong et al. [20] divided it into several periods and analyzed the user's preference distribution in each period and quantified their preferences. Li et al. [21] split user preferences to stages over time and proposed the cross-domain CF framework. The experiments proved that the algorithm not only improves the recommendation prediction accuracy but also solves the problem of user interest drift.

3. Description of RIT-UA Algorithm

In the context of relatively sparse data, from the perspective of solving the problem of user interest drift, this paper proposes the RIT-UA algorithm on the basis of the traditional similarity calculation, with the introduction of factors (such as the user attribute characteristics and time decay of rating) which influence user's rating behaviors. The RIT-UA algorithm consists of two parts: one is the similarities of rating-interest, and the other part is the similarities of the user attributes.

3.1. The Similarities of Rating-Interest. The similarities of rating-interest are composed of rating similarity and interest similarity, mainly considering two aspects: users' preference for items and user's rating habits. Meanwhile, based on the

Table 1: User-item rating matrix.

	Item 1	Item 2	Item 3	Item 4
User 1	4	5	-	1
User 2	-	2	3	2
User 3	1	-	3	4
User 4	-	2	-	3

two aspects, the effect of time decay of rating is introduced and the confidence coefficient between users is also introduced with the combination of the fluctuation factor proposed in literature [4]. In the end, the similarities of rating-interest between users are obtained. The whole process is described as follows.

3.1.1. Rating Similarity. In the field of e-commerce systems, Rating or Voting is generally used to obtain the user's direct preference for items. Assuming that the degree of user's preference for items is classified as 5 levels, which is {adore, love, like, dissatisfied, and dislike}, and the corresponding grades are {5, 4, 3, 2, 1}. Consequently, the results will produce a rating matrix. The rating preference matrix of user-item can be shown in Table 1.

Table 1 is a rating matrix of user-item. In the rating matrix, when the ratings of two users are closer, it indicates that their preferences are similar. When the ratings of two users are the same, it implies that the users share the same preferences. If there is big gap between ratings, then it means that the two users have opposite preferences. Therefore, in order to describe the nonlinear correlation of the similarity between users' ratings, the paper constructs sigmoid function to express the similarity of user ratings based on the literature [22], in which sigmoid function is put forward as the expression of the similarity. In this paper, the sigmoid function is also used to represent the similarity between users. The equation is shown below:

$$\text{sim}(u, v, i)_{\text{rate}} = 2 \cdot \left(1 - \frac{1}{1 + \exp\left(-\left|r_{ui} - r_{vi}\right|\right)}\right). \quad (1)$$

Equation (1) represents the similarity between the ratings for item i from user u and user v. r_{ui} represents the ratings for item i from user u and r_{vi} represents the ratings for item i from user v.

3.1.2. Interest Similarity. Every user has their own rating habits. For instance, some users who do not stick to rifles always tend to give a high score, while some rigorous users who pay much attention to details is likely to give a low score. Because they are more strict with the score, they do not give high scores easily. Hence, the description of user habits is helpful to improve the prediction accuracy. For the user rating habits and the inherent attributes of the item, Koren [17] used an equation to define them, as shown in equation (2), that is, regarding user's own rating habits as a factor having an impact

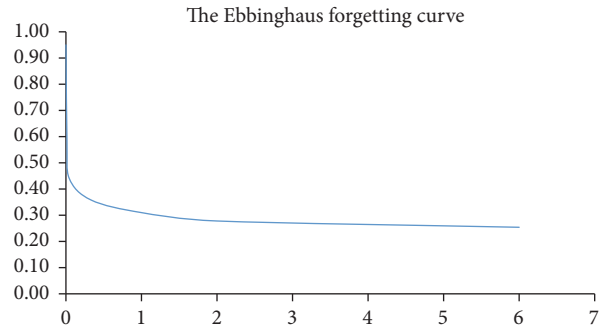


Figure 1: Changes of Ebbinghaus forgetting curve.

on rating. In (2) b_u stands for the user's own rating habits, while b_i stands for the user's rating for item i.

$$b_{ui} = \mu + b_u + b_i. \quad (2)$$

Therefore, within the range of rating for items, when a user tends to score highly and likes an object, he/she usually gives a high score for it. However, even though the user does not like the object, he/she will not give a low score and vice versa. Therefore, according to the average score given by the user for an item, his/her interest and preference of rating habits can be showed. Similarly, based on literature [22], which proposed sigmoid function as the expression of the similarity, the paper also constructs sigmoid function to express the similarity of user interests, shown in

$$\text{sim}(u, v, i)_{\text{interest}} = 1 - \frac{1}{1 + \exp\left(-\left(r_{ui} - \overline{r}_u\right)\left(r_{vi} - \overline{r}_v\right)\right)}. \quad (3)$$

Equation (3) represents the similarity of the interest of user u and user v on item i. Then, combining the rating similarity and interest similarity between users, we get a computational equation, shown as

$$\text{sim}(u, v)'_{\text{score}} = \text{sim}(u, v, i)_{\text{rate}} + \text{sim}(u, v, i)_{\text{interest}}. \quad (4)$$

3.1.3. Time Factor. Generally speaking, treating user behaviors that occurred at various time equally leads to the shortage of effective quantitative analysis. Time factor shows the degree of changing tendency of user interest drift. The closer the rating information to the present time, the better recommendation effects it has and vice versa. Based on this, some studies used linear and nonlinear functions to quantify the rating behaviors over time.

In the literature [23], in order to solve the difficult problem of tracking the changes of user interest, the Ebbinghaus forgetting curve is put forward for the research of user interest fitting. Changes of Ebbinghaus forgetting curve is shown in Figure 1. Based on the literature [23], combined with the trend of Ebbinghaus forgetting curve this paper uses the following function to describe the trend of user interest drift, that is, draw the impact direction of the time factor, as shown in

$$f(\Delta t) = \frac{2e^{-\alpha \Delta t}}{1 + e^{-\alpha \Delta t}} \in (0, 1]. \quad (5)$$

Δt represents the time difference between users' rating on item I, which is the parameter, and in this paper, we set it as 0.005. After taking time-effect into account, therefore, the new computational equation for similarities of user rating-interest arrives:

$$\mathrm{sim}(u,v)''_{\mathrm{score}} = \frac{1}{|I_{uv}|} \sum_{i \in I_{uv}} \left(\mathrm{sim}(u,v)'_{\mathrm{score}} \cdot f(\Delta t) \right). \quad (6)$$

$|I_{uv}|$ represents the number of items corated by user u and user v.

3.1.4. Confidence between Users. When the user data is extremely sparse and the number of corated items is very small, there is a large fortuitous factor in the similarity calculation. Li et al. [4] eliminate this effect by using the fluctuation factor. Based on this, the paper introduces the number of corated items to adjust the weight of similarity through nature exponential, shown as

$$\mathrm{confident}(u,v) = \exp\left(\frac{|I_u \cap I_v|}{\max(|I_u \cap I_w|)} - 1 \right). \quad (7)$$

Equation (7) represents the confidence coefficient between user u and user v, stands for the item rated by user u, stands for the item rated by user v, shows the corated items of user u and user v, represents the corated item between user u and the nearest neighbor, and stands for the nearest neighbor set.

After taking confidence coefficient into account, the adjusted equation to calculate the similarity of user rating-interest arrives:

$$\mathrm{sim}(u,v)_{\mathrm{score}} = \mathrm{sim}(u,v)''_{\mathrm{score}} \cdot \mathrm{confident}(u,v). \quad (8)$$

3.2. The Similarity of User Attributes. Considering the similarity of user attributes, on the one hand it can improve the accuracy of prediction, and on the other hand it can solve the problem of new user's cold start; that is, when there is no other available rating data, data of user attribute features can be used to build models and give recommendations. As for the description about the similarity of user attributes, literature [20] divided the user attributes into numerical attributes and name attributes and defined and expressed them, respectively. From the perspective of being easy to understand and implement, this paper defines the similarity of user attributes as follows.

For single user attribute, it is expressed as $\mathrm{sim}(u,v,i)_{\mathrm{attr}} = 1/0$.

It indicates that when user u and user v share the same attribute i, the value is 1; otherwise the value is 0.

$$\mathrm{sim}_{\mathrm{attr}}(u,v) = \sum_{i \in \mathrm{Attr}} w \cdot \mathrm{sim}(u,v,i)_{\mathrm{attr}}. \quad (9)$$

In (9) w is the value of feature weight of user attribute i. In order to obtain all weight values of each feature attribute, this paper chooses the feature selection algorithm of Random Forest to calculate the importance degree of each user attributes feature and generates a rank of it. Then we conduct experiments according to the rank and acquire the relative importance weight value of each attribute further.

```
Algorithm 1 RIT-UA similarity calculation
Input:
    Testset
Algorithm
(1) For user in Testset do:
(2)    For item in Testset[user] do:
(3)      //get co-rated items
(4)      Users: getCorateditemsUserset(item)
(5)      //get the similarity between user and Users
(6)      calculateRituaSimilarity(Users, user)
(7)      //According similarity select neighbors
(8)      getTonKNeighbors(K)
(9)      //calculate predicted rating
(10)     rating: getRating(Neighbors)
(11)  end for
(12) end for
```

Algorithm 1: Description of RIT-UA algorithm.

3.3. Similarity Calculation Based on RIT-UA. Sections 3.1 and 3.2 consider the similarity of rating-interest and the similarity of user attributes, respectively; hence we carry out weighted combination for the two and get a new computational equation of similarity:

$$\mathrm{sim}(u,v) = \alpha \cdot \mathrm{sim}_{\mathrm{score}}(u,v) + \beta \cdot \mathrm{sim}_{\mathrm{attr}}(u,v). \quad (10)$$

In (10), $\beta = 1 - \alpha$. After the computational equation of similarity is obtained, we get the prediction equation of user to item, shown as

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in U} \mathrm{sim}(u,v) \cdot (r_{vi} - \bar{r}_v)}{\sum_{v \in U} |\mathrm{sim}(u,v)|}. \quad (11)$$

$\bar{r}_u$ and $\bar{r}_v$ mean the average scores of user u and user v, respectively, and U stands for the neighbor set of users u.

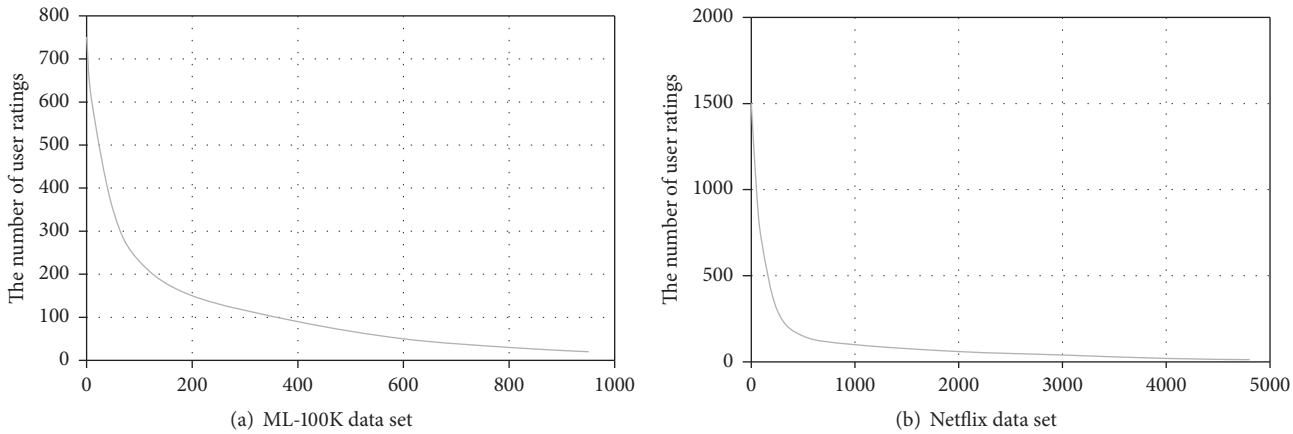
The description of RIT-UA similarity algorithm is in Algorithm 1.

Therefore, from the description in Algorithm 1 we can see that the time complexity of operating RIT-UA algorithm is $O(m * n)$, where m means the number of users and n means the number of items.

4. Description of RIT-UA Algorithm

4.1. Experimental Data Sets. Taking into account the openness and authority of data sets, at the same time, our simulation experiment is based on the scoring matrix, so we chose two data sets, namely, Movielens-100k and Netflix, to carry out experimental analysis and comparison. The process is shown as follows.

4.1.1. Movielens-100k Data Set. The data set is a film rating data set provided by the GroupLens Research. The data set contains 100,000 ratings from 943 users for 1682 movies, where each user has rated 20 movies at least, and the rating interval is {1–5} which is shown as Table 2. Meanwhile, the sparseness of the data set is $1 - 100000/(943 * 1682) = 93.7\%$.



(a) ML-100K data set (b) Netflix data set

Figure 2: Changing tendency of the number of user-rated items (descending order).

Table 2: User-item rating matrix.

	Movielens-100k	Netflix
Users	943	4861
Items	1682	5080
Ratings	100000	387939
Rating scale	{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}
Sparseness of data	93.7%	98.4%

Figure 2(a) shows the number of items rated by users on the ML-100k data set in a descending order. From the figure, we can see that the number of items rated by many users is less than 100. In order to test the performance of the algorithm, the data set is divided into two parts: 80% as the training set and 20% as the test set.

In ML-100k data set, there are only 4 attributes about users' attribute feature: gender, age, occupation, and zip code.

4.1.2. Netflix Data Set. Netflix data set is a section of the original Netflix Game data. After the proper data cleaning, the data set contains 387,939 ratings from 4861 users for 5080 objects, where each user has rated 20 objects at least, and the rating interval is {1–5} which is shown as Table 2.

The sparseness of the data set is $1-387939/(4861*5080) = 98.4\%$, and Figure 2(b) shows the number of items rated by users on the ML-100k data set in a descending order. From the figure, we can see that the number of items rated by a large number of users is less than 100. Similarly, in order to test the performance of the algorithm, the data set is divided into two parts: 80% as the training set and 20% as the test set.

In the process of cleaning the Netflix data set, since there is no user attribute feature data in it, according to the features of the user attribute data of ML-100k, this paper randomly generates data of three user attributes in Netflix through the simulation experiment: gender, age, and occupation. The range of age attribute is {10–65}, the occupation attribute has 20 occupations with the range {0–19}, and the value of gender is given within the range {0–2}. Because of the high sparsity of our data sets, we use resource scheduling and processing methods for sparse data [24, 25].

4.2. Experiment Evaluation Quantity. Generally speaking, there are evaluation quantities such as MAE (mean absolute error) and RMSE (root mean squared error) in the experimental evaluation about prediction precision in recommender systems. After comparison, RMSE (root mean squared error) is used as the evaluation quantity in this paper. The equation is

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i\in \mathrm{Test}} \left(r_{ui} - \hat{r}_{ui}\right)}{\left|N_{\mathrm{Test}}\right|}}. \tag{12}$$

$|N_{\mathrm{Test}}|$ represents the size of test data set and refers to the real rating value while referring to the predicted rating value. The smaller the value shown by RMSE, the higher the predicted precision; that is, the smaller the value, the closer the prediction.

4.3. Experimental Process and Analysis

4.3.1. Experiment 1: Experimental Analysis of the Weight Value of User Attribute Feature. From (9) we can see that, in order to obtain the weighted value of each user attribute feature, the Random Forest algorithm is chosen in this paper.

Random forests are an ensemble learning method that can analyze the complicated interactive feature data, even under the influence of certain data noise it is very robust, and it is very efficient in feature learning and analysis. Its variable importance measure can be a feature selection tool for high dimensional data. In recent years, it has been widely used in various kinds of prediction, feature selection, and outlier detection [26].

Therefore, we obtain the weight value of each user attribute feature with Random Forest algorithm on ML-100k and Netflix data set. The experimental results are shown in Figures 3 and 4.

On ML-100k data set, from Figure 3 we can see that among the 4 attributes (age, gender, occupation, and zip code) gender is the most important, indicating that gender attribute exerts a significant role in recommendation and the user rating is more similar when it relates to this attribute. Compared to the gender attribute, zip code attribute exerts

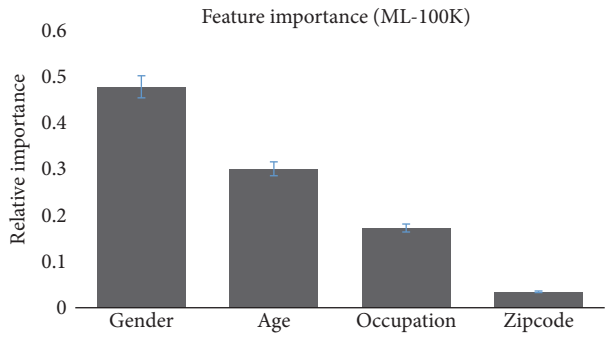


Figure 3: Ranking of the weight value of user attributes feature (ML-100k).

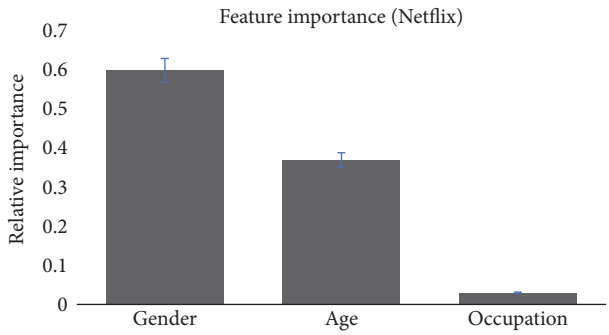


Figure 4: Ranking of the weight value of user attributes feature (Netflix).

a relatively low role in recommendation so its weight value is low correspondingly. But the other two attributes age and occupation show relatively medium influence of feature weight, as the experiment implies whose weight value is about 0.284 and 0.186, respectively.

The illustration parts of Figures 3 and 4 show the domain of walker of possible weight values for each feature. For the Netflix data set, the gender and age attributes have very obvious effects in recommendation, and the overall importance rank of the weight value is similar to ML-100k.

In order to test the relative optimal weight values of every and each attribute of (age, gender, occupation, and zip code) and (age, gender, and occupation) on the ML-100k and Netflix data sets, we carry out several sets of comparative experiments in this paper, and experimental results are shown in Figures 5 and 6. From Figures 5 and 6 we can see that on ML-100k data set when "age, gender, occupation, and zip code" are given the values "0.3, 0.3, 0.25, and 0.15," respectively, we get better experimental results. As for Netflix data set, when "age, gender, and occupation" are given the values "0.5, 0.4, and 0.1," respectively, we get better experimental results. Therefore, the values above will be used in the following experiments.

4.3.2. Experiment 2: Experimental Analysis of the Weight Value of Alpha and Beta. According to (10), in order to get the values of α and β which will generate relatively good experimental results, we used the RIT-UA algorithm to carry

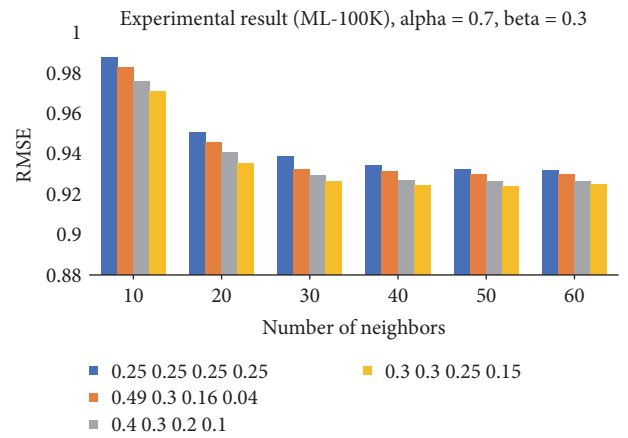


Figure 5: Experiment comparison of weight values of different user attributes (ML-100k).

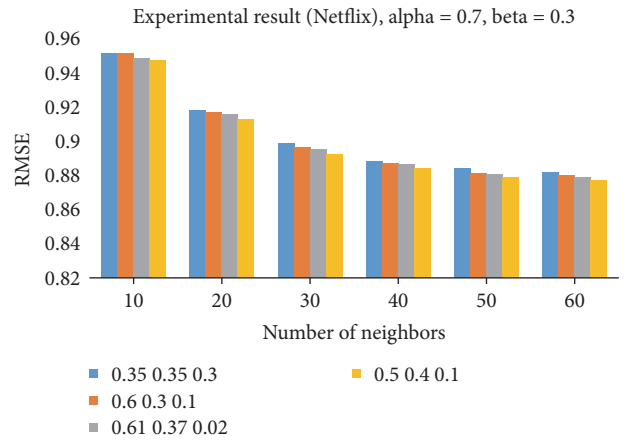


Figure 6: Experiment comparison of weight values of different user attributes (Netflix).

out the following groups of experiments based on ML-100k and Netflix data sets. The experimental results are shown in Figures 7 and 8.

From results shown by Figures 7 and 8 we can see that, for ML-100k data set, when $\alpha = 0.75$ and $\beta = 0.25$, we get relatively better experimental results. And, for Netflix data set, when $\alpha = 0.7$ and $\beta = 0.3$, the relatively better experimental results are obtained.

4.3.3. Experiment 3: Experimental Analysis of the Comparison with Other Similarity Measures. In order to verify the validity of the algorithm proposed in this paper, we compare it with other similarity measures, including the Pearson similarity, the adjusted cosine similarity (Acosine), the PIP [27] similarity, and the NHSM [28] similarity on the ML-100k and Netflix data sets. The experimental results are shown in Figures 9 and 10, respectively.

From Figure 9 we know that, on ML-100k data set, the overall experimental results show that, with the increase of neighbors, the algorithm of this paper outperforms others

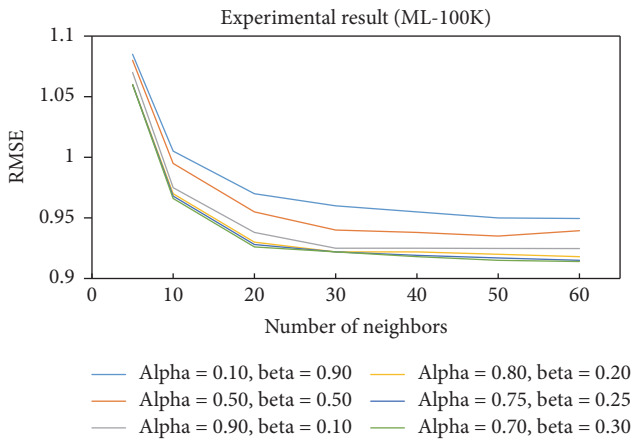


Figure 7: Experimental results with different alpha and beta (ML-100k).

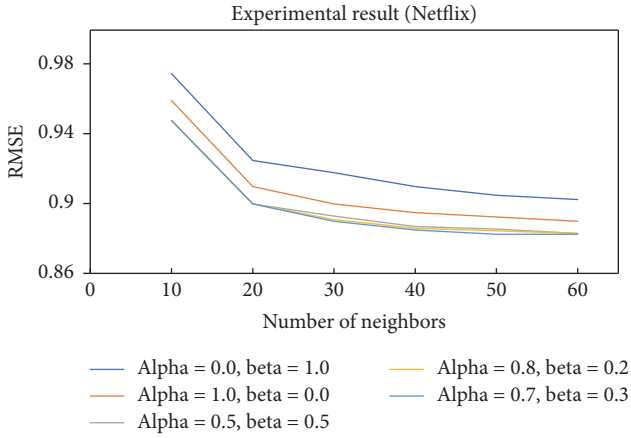


Figure 8: Experimental results with different alpha and beta (Netflix).

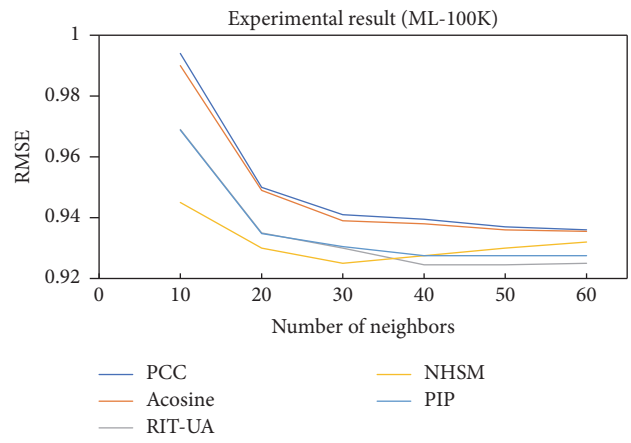


Figure 9: Experimental comparison of different similarity algorithms (ML-100k).

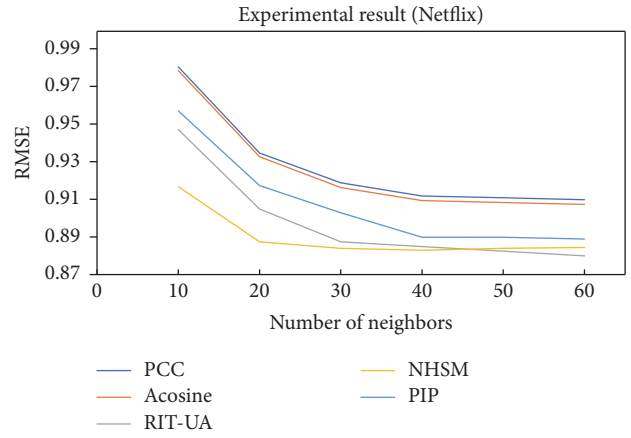


Figure 10: Experimental comparison of different similarity algorithms (Netflix).

gradually. At the beginning stage when the number of neighbors is within [10, 30], the results of the algorithm proposed by this paper are close to those of PIP but slightly better than that of PIP in later period. The experimental results of NHSM are good when the number of neighbors is within [10, 30] but worse in later period. The experimental results of PCC and Acosine are worse than other algorithms. On Netflix data set, from Figure 10 we know that the algorithm proposed in this paper outperforms other algorithms gradually with the increase of neighbors. NHSM outperforms others when the number of neighbors is within [10, 40] but performs not so well as the algorithm proposed by this paper later.

4.3.4. Experiment 4: Comparison of Precision on Data Sets of Different Sizes. Based on ML-100k data set, the paper chooses 20%, 40%, 60%, and 80% of the data set, respectively. Neighbors $k = 20$ as a prerequisite, and we verify the comparison of precision of different algorithms on data set of various sizes. Fivefold cross validation is used to get the average value of experimental results, which is shown as Figure 11.

From Figure 11 we can see that the proposed algorithm produces better and stable results on varied sizes of ML-100k data sets, indicating that, in the case of sparse data, the proposed algorithm has higher identification degree. As for the other three algorithms, performance of PIP algorithm is relatively stable, and RMSE value is relatively low. However, for NHSM algorithm, RMSE value is higher when the data set is relatively sparse, while, with the sizes of the data set increase, the NHSM algorithm performs better and becomes more stable.

5. Conclusion

Aiming at some problems in traditional similarity calculation, this paper proposes a new similarity calculation model. The model describes and expresses aspects such as user rating preference, user rating habits, and time factor. Furthermore, user attributes feature is taken into account for its influence on user ratings, and the role of each attribute feature played in recommendation is studied. Then Random Forests algorithm is used to calculate the weight value of each attribute. The final

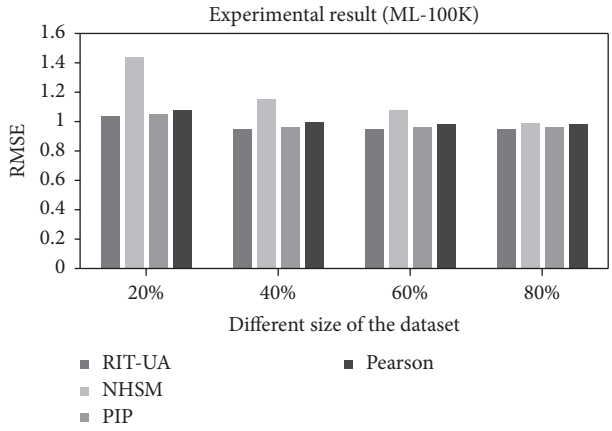


Figure 11: Comparison of results produced by different algorithms on data sets of different sizes (ML-100k).

experimental results show that, compared to other similarity measures, the approach proposed in this paper improves the recommendation precision significantly, and even in the case of sparse data it still shows better experimental results. The deficiency of experiments is that since the user attribute data is relatively small in data set, there is no obvious difference when calculating the feature weight value of user attributes, as the part of user attributes data is private and not easy to obtain, which inevitably cast a shadow on the experiments.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This project is supported by the Fundamental Research Funds for the Central Universities of Central South University with no. 2017zzts623 and Hunan Provincial 2011 Collaborative Innovation Center for Development and Utilization of the Financial and Economic Big Data Property.

References

[1] H. Luo, C. Niu, R. Shen, and C. Ullrich, "A collaborative filtering framework based on both local user similarity and global user similarity," *Machine Learning*, vol. 72, no. 3, pp. 231–245, 2008.

[2] D. Anand and K. K. Bharadwaj, "Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5101–5109, 2011.

[3] A. R. S. Lopes, R. B. C. Prudencio, and B. L. D. Bezerra, "A collaborative filtering framework based on local and global similarities with similarity tie-breaking criteria," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 2887–2893, July 2014.

[4] H. Li, G. Wang, and M. Gao, "A novel similarity calculation for collaborative filtering," in *Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition*, pp. 38–43, IEEE, July 2013.

[5] J. Shen, Y. Wei, and Y. Yang, "Collaborative filtering recommendation algorithm based on two stages of similarity learning and its optimization," in *Proceedings of the 13th IFAC Symposium on Large Scale Complex Systems: Theory and Applications*, pp. 335–340, July 2013.

[6] L. Gao and M. Huang, "A collaborative filtering recommendation algorithm with time adjusting based on attribute center of gravity model," in *Proceedings of the 12th Web Information System and Application Conference*, pp. 197–200, September 2015.

[7] X. Luo, J. Deng, J. Liu, W. Wang, X. Ban, and J. Wang, "A quantized kernel least mean square scheme with entropy-guided learning for intelligent data analysis," *China Communications*, vol. 14, no. 7, pp. 1–10, 2017.

[8] J. Beel, S. Langer, A. Nürnberger et al., "The impact of demographics (age and gender) and other user-characteristics on evaluating recommender systems," in *Research and Advanced Technology for Digital Libraries*, pp. 396–400, Springer, Berlin, Germany, 2013.

[9] Y. Wang, S. C.-F. Chan, and G. Ngai, "Applicability of demographic recommender system to tourist attractions: a case study on trip advisor," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops*, pp. 97–101, December 2012.

[10] W. Zhao, R. Lun, C. Gordon et al., "A human-centered activity tracking system: toward a healthier workplace," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, pp. 343–355, 2017.

[11] M. Y. Al-Shamri, "User profiling approaches for demographic recommender systems," *Knowledge-Based Systems*, vol. 100, pp. 175–187, 2016.

[12] E. B. Santos, M. Garcia Manzato, and R. Goularte, "Evaluating the impact of demographic data on a hybrid recommender model," *IADIS International Journal on WWW/Internet*, vol. 12, no. 2, pp. 149–167, 2014.

[13] T. Chen and L. He, "Collaborative filtering based on demographic attribute vector," in *Proceedings of the International Conference on Future Computer and Communication*, pp. 225–229, IEEE Computer Society, June 2009.

[14] X. Luo, H. Luo, and X. Chang, "Online optimization of collaborative web service QoS prediction based on approximate dynamic programming," in *Proceedings of the International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI '14)*, pp. 80–83, IEEE, Beijing, China, October 2014.

[15] X. Luo, J. Liu, D. D. Zhang, and X. Chang, "A large-scale web QoS prediction scheme for the Industrial Internet of Things based on a kernel machine learning algorithm," *Computer Networks*, vol. 101, pp. 81–89, 2016.

[16] L. Y. Dou and X. H. Wang, "A collaborative filtering recommendation algorithm based on the context of time and tags," *Journal of Taiyuan University of Technology*, no. 6, 2015.

[17] Y. Koren, "Collaborative filtering with temporal dynamics," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pp. 447–456, Paris, France, June 2009.

[18] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering," in *Proceedings of the 4th ACM Recommender Systems Conference (RecSys '10)*, pp. 79–86, ACM, Barcelona,Spain, September 2010.

[19] L. Xiong, X. Chen, T. K. Huang et al., "Temporal collaborative filtering with bayesian probabilistic tensor factorization," in *Proceedings of the Siam International Conference on Data Mining (SDM '10)*, pp. 211–222, Columbus, Ohio, USA, April-May 2010.

[20] H. G. Rong, S. X. Huo, C. H. Hu et al., "User similarity-based collaborative filtering recommendation algorithm," *Journal on Communications*, vol. 35, no. 2, pp. 16–24, 2014.

[21] B. Li, X. Zhu, R. Li et al., "Cross-domain collaborative filtering over time," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '11)*, pp. 2293–2298, Barcelona, Spain, July 2011.

[22] M. Jamali and M. Ester, "TrustWalker: a random walk model for combining trust-based and item-based recommendation," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pp. 397–405, July 2009.

[23] H. Yu and Z. Y. Li, "A collaborative filtering recommendation algorithm based on forgetting curve," *Journal of Nanjing University (Natural Sciences)*, vol. 46, no. 5, pp. 520–527, 2010.

[24] W. Wei, X. Fan, H. Song, X. Fan, and J. Yang, "Imperfect information dynamic stackelberg game based resource allocation using hidden markov for cloud computing," *IEEE Transactions on Services Computing*, 2016.

[25] T. Li, Y. Liu, L. Gao, and A. Liu, "A cooperative-based model for smart-sensing tasks in fog computing," *IEEE Access*, vol. 5, pp. 21296–21311, 2017.

[26] D.-J. Yao, J. Yang, and X.-J. Zhan, "Feature selection algorithm based on random forest," *Journal of Jilin University (Engineering and Technology Edition)*, vol. 44, no. 1, pp. 137–141, 2014.

[27] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Information Sciences*, vol. 178, no. 1, pp. 37–51, 2008.

[28] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," *Knowledge-Based Systems*, vol. 56, pp. 156–166, 2014.

Hindawi
Scientific Programming
Volume 2017, Article ID 3610378, 10 pages
https://doi.org/10.1155/2017/3610378

Research Article

A Robust Text Classifier Based on Denoising Deep Neural Network in the Analysis of Big Data

Wulamu Aziguli,[1,2] Yuanyu Zhang,[1,2] Yonghong Xie,[1,2] Dezheng Zhang,[1,2] Xiong Luo,[1,2,3] Chunmiao Li,[1,2] and Yao Zhang[4]

[1]*School of Computer and Communication Engineering, University of Science and Technology Beijing (USTB), Beijing 100083, China*
[2]*Beijing Engineering Research Center of Industrial Spectrum Imaging, Beijing 100083, China*
[3]*Key Laboratory of Geological Information Technology, Ministry of Land and Resources, Beijing 100037, China*
[4]*Tandon School of Engineering, New York University, Brooklyn, NY 11201, USA*

Correspondence should be addressed to Yonghong Xie; xieyh@ustb.edu.cn, Dezheng Zhang; zdzchina@ustb.edu.cn, and Xiong Luo; xluo@ustb.edu.cn

Received 25 August 2017; Accepted 17 October 2017; Published 27 November 2017

Academic Editor: Anfeng Liu

Text classification has always been an interesting issue in the research area of natural language processing (NLP). While entering the era of big data, a good text classifier is critical to achieving NLP for scientific big data analytics. With the ever-increasing size of text data, it has posed important challenges in developing effective algorithm for text classification. Given the success of deep neural network (DNN) in analyzing big data, this article proposes a novel text classifier using DNN, in an effort to improve the computational performance of addressing big text data with hybrid outliers. Specifically, through the use of denoising autoencoder (DAE) and restricted Boltzmann machine (RBM), our proposed method, named denoising deep neural network (DDNN), is able to achieve significant improvement with better performance of antinoise and feature extraction, compared to the traditional text classification algorithms. The simulations on benchmark datasets verify the effectiveness and robustness of our proposed text classifier.

1. Introduction

While entering the era of big data with the development of information technology and the Internet, the amount of data is getting geometric growth. We are entering information overload era. The issue that people are facing is no longer how to get information, but how to extract useful information quickly and efficiently from massive amount of data. Therefore, how to effectively manage and filter information has always been an important research area in engineering and science fields.

With the rapid increase of the amount of data, information representation is also diversified, mainly including text, sound, and image. Compared with sound and image, text data uses less network resources and is easier to be uploaded and downloaded. Since other forms of information can be also expressed by text, text has become the main carrier of information and always occupies a leading position in the network resources.

Traditionally, it is time-consuming and difficult to achieve the desired results of text processing, and it can not adapt to the demand of information society for explosive growth of digital information. Hence, effectively obtaining information in accordance with the user feedback can help users to get the information quickly and accurately. Then, text classification becomes a critical technology to achieve free human-machine interaction and contribute to artificial intelligence. It can address the messy information issue to a large extent, so that users can locate the information accurately.

1.1. Text Classification. The purpose of text classification is to assign large amounts of text to one or more categories based on the subject, content, or attributes of the document. The methods of text classification are divided into two categories, including rules-based and statistical classification methods

[1, 2]. Among them, the rules-based classification methods need more knowledge and rules base in this field. However, the development of rules and the difficulties of updating them make the application of this method relatively narrow and suitable for only a specific field. Statistical learning methods are usually based on a statistic or some kinds of statistical knowledge; these methods establish learning parameters of the corresponding data model through the sample statistics and calculation on the train set and then conduct the training of the classifier. In the test stage, the categories of the samples could be predicted according to these parameters.

Recently, a large number of statistical machine learning methods are applied to the text classification system. The application of the earliest machine learning method is naive Bayes (NB) [3, 4]. Subsequently, almost all the important machine learning algorithms have been applied to the field of text classification, for example, K nearest neighbor (KNN), neural network (NN), support vector machine (SVM), decision tree, kernel learning, and some others [5–10]. SVM uses the shallow linear model to separate the objective. In low dimensional space, when different types of data vectors can not be divided, SVM will map it to a high dimensional space through kernel function and finds the optimal hyperplane. In addition, NB, linear classification, decision tree, KNN, and other methods are relatively weak, but their models are simple and efficient; then those methods are accordingly improved.

But these models are shallow machine learning methods. Although they have also been proven to be able to efficiently address some of the issues in the case of simple or multiple restrictions, when facing complex practical problems, for example, biomedical multiclass text classification, the data is noisy and dataset distribution is uneven classification and shallow machine learning model and generalization ability of integrated classifier method will be unsatisfactory. Therefore, the exploration of some other new methods, for example, deep learning method, is necessary.

1.2. Deep Learning. With the success of deep learning methods [11, 12], some other improvement for NN, for example, deep belief network (DBN) [13], has been developed. Here, DBN is designed on the basis of the cascaded restricted Boltzmann machine (RBM) [14] learning algorithm, through unsupervised greedy layer pretraining strategy combining the supervision of fine-tuning training methods. It can tackle the problem of complex deep learning model optimization, so that the deep neural network (DNN) has witnessed the rapid advancements.

Meanwhile, DNN has been applied to many learning tasks, for example, voice and image recognitions [15]. For example, since 2011, Microsoft and Google's speech recognition research team achieved a voice recognition error rate reduction of 20%–30% using DNN model, stepping forward in the field of speech recognition in the past decades. In 2012, DNN technology in the ImageNet [15] evaluation task (image recognition field) improved the error rate from 26% to 15% [16].

Moreover, the automatic encoder (AE) as a DNN reproduces the input signal [17, 18]. Its main principle is that there is a given input; it first encodes the input signal using an encoder and then decodes the encoded signal using a decoder, while achieving the minimum reconstruction error by constantly adjusting the parameters of encoder and decoder [19]. Additionally, there are some improvements to AE, for example, sparse AE and denoising AE [17, 18]. The performance of some machine learning algorithms could be further improved through the use of those AEs [20].

Recently, deep learning methods have a significant impact on the field of natural language processing (NLP) [11, 21].

1.3. Status Analysis. Due to the complex feature of large text data, and different effects of noise, the performance is not satisfactory when dealing with large dataset using traditional text classification algorithms.

More recently, deep learning has been applied to a series of classification issues with multiple modes successfully. Then, the user can effectively extract the complex semantic relations of the text by using deep learning-based methods [11, 22]. With the popularity of deep learning algorithms, DNN has some advantages in dealing with large-scale dataset. In this article, motivated by DNN, the denoising deep neural network (DDNN) is designed and the feature extraction is conducted by using this model.

For the shallow text representation (feature selection), there is a problem of missing semantics. For the deep text representation of the model based on the linear calculation, the selection of the threshold is added to the classifier training, which actually destroys the self-taught learning ability of the text. Meanwhile, for text classification of multilabel and multicategory, there is also a problem of ignoring label dependencies and lack of generalizing ability. To cope with the above problems, some improvements are achieved through deep learning methods. For example, a two-layer replicated softmax model (RSM) was proposed in [23], which is better than latent Dirichlet allocation (LDA), that is, a semantically consistent topic model [24]. However, the model is designed using weighted sharing technique and there are only two layers. In the process of dimension reduction, the missing information of documents is relatively larger, and the ability of noise handling is poor, resulting in little difference between different documents using the model.

In order to avoid such limitations and develop a better approach, this article proposes a DDNN model through the combination of some state-of-the-art deep learning methods. Specifically, in our model, the data is denoised with the help of denoising autoencoder (DAE), and then the feature of the text is extracted effectively using RBM. Compared with those traditional text classification algorithms, our proposed algorithm can achieve significant improvement with better performance of antinoise and feature extraction, due to the efficient learning ability of hybrid deep learning methods used in this model.

The reminder of this article is organized as follows. In Section 2, we give a technique analysis for DAE [25] and RBM [26]. Then, our proposed text classifier is presented in Section 3, where more attention is paid for the implementation of DDNN. Section 4 provides some simulation results and discussions. Finally, the conclusion is given in Section 5.

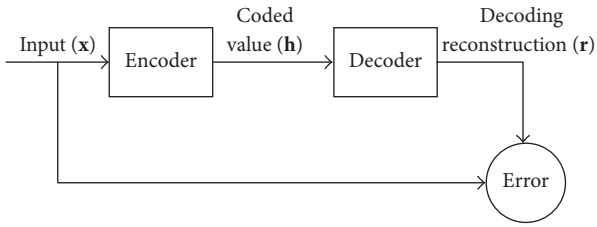


Figure 1: Schematic diagram of automatic encoder model.

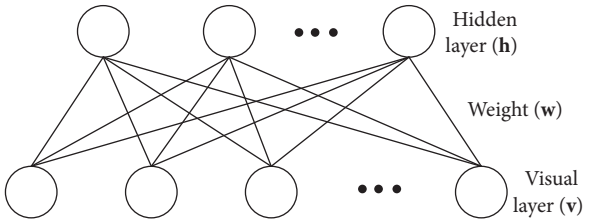


Figure 2: Schematic diagram of restricted Boltzmann machine.

2. Background

In this article, we use two kinds of state-of-the-art deep learning models, that is, DAE and RBM [25, 26].

2.1. Denoising Autoencoder (DAE). Generally, the structure of AE [27] is shown in Figure 1. Here, the whole system consists of two networks, that is, encoder and decoder. Its purpose is to make the reconstruction layer output as similar to the input as possible. The coding network will code and calculate the input **x** and then reconstruct the result **h** to **r** by the decoder. And denoising automatic coding is developed according to the automatic coding, it will learn a more robust representation of the input signal and has stronger generalization ability than ordinary encoders by adding noise to the training data.

2.2. Restricted Boltzmann Machine (RBM). As shown in Figure 2, RBM network has two layers [28, 29]. Here, the first layer is the visual layer (**v**), also called the input layer, which consists of m visible nodes. And the second layer is the hidden layer (**h**), that is, the feature extraction layer, and it consists of n hidden nodes. If v is known, then $P(h/v) = P(h_1/v) \cdots P(h_n/v)$ and all hidden nodes are conditional independent. Similarly, all the visible nodes are also conditional independent when the hidden layer **h** is known, the nodes within the layer are not connected, and the nodes from different layers are fully connected.

3. The Proposed Text Classifier

3.1. Denoising Deep Neural Network (DDNN)

3.1.1. Framework. Here, a DDNN is designed using DAE and RBM, which can effectively reduce the noise while extracting the feature.

The input of the DDNN model is a vector with fixed dimension. Firstly, we conduct the training by the denoising module composed of two layers, named DAE1 and DAE2, using unsupervised training methods. Here, only one of them is trained each time, and each training can minimize the reconstruction error for the input data, that is, the output of the previous layer. Because we can calculate the encoder or its potential expression based on the previous layer k, so the $(k+1)$th layer could be processed directly using the output of the kth layer, until all the denoising layers are trained.

The operation of this model is shown in Figure 3.

After being processed through the denoising layer, the data enters the portion of RBM, which can further extract the feature that is different from the denoising autocoder layer. The feature extracted after this part will be more representative and essential. Figure 4 is the diagram for the RBM feature extraction.

This part is constructed by stacking two layers of RBM. Training can be conducted by training RBM from low to high as follows.

(1) The input of bottom RBM is the output of the denoising layer.

(2) The feature extracted from the bottom RBM is taken as the input of the top RBM.

Because RBM can be trained quickly by contrastive divergence (CD) learning algorithm [30], this training framework avoids the high complexity calculation of directly getting a deep network with one training by dividing it into multiple RBMs training. After this training, the initial parameter values of some pretraining models are obtained. Then, a backpropagation (BP) NN is initialized using these parameters; the network parameters are fine-tuned by the traditional global learning algorithm using the dataset with tags. Thus, the function can converge to the global optimal point.

The reason for choosing DAE here is that, in the process of text classification, data will be inevitably mixed into different types and intensity of noise, which tends to affect the training of the model, resulting in deterioration of the final classification performance. DAE is a preliminary extraction of the original features, and its learning criteria is noise reduction. In the pretraining stage, adding a variety of different strength and different types of noise signals to the original input signal can make the encoding process obtain better stability and robustness. It is shown in Figure 5.

Moreover, the reason for choosing RBM is that RBM is characterized by the fact that it can simulate the discrete distribution of arbitrary samples and it is very suitable for feature expression when the number of hidden layer units is sufficient.

3.1.2. Implementation. The DDNN model consists of four layers, that is, DAE1, DAE2, RBM1, and RBM2. The layer **v** is both visual layer and the input layer of the DDNN model. Each document in this article is represented by a fixed dimension vector, where W_1, W_2, W_3, and W_4 represent the connection weight between the layers, respectively. In addition, h_1, h_2, h_3, and h_4 represent each hidden layer corresponding to the output layers DAE1, DAE2, RBM1, and

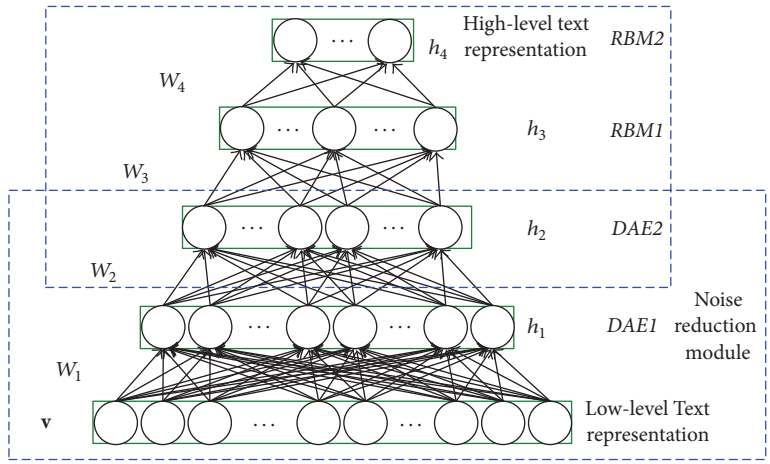


Figure 3: Schematic diagram of denoising deep neural network.

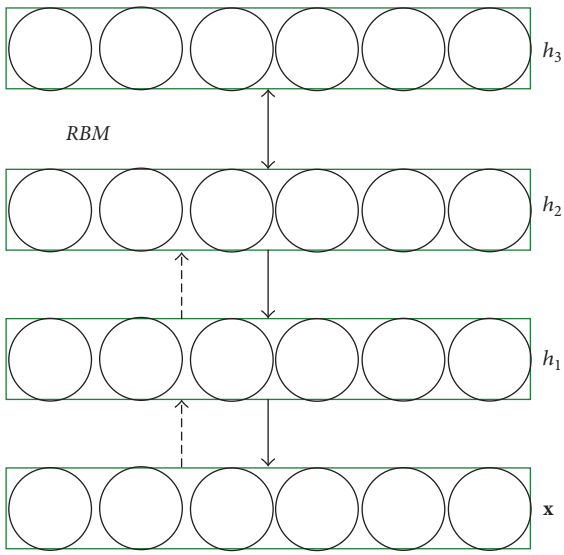


Figure 4: Illustration of feature extraction in RBM.

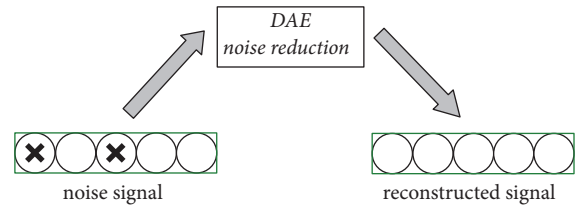


Figure 5: Noise reduction with DAE.

RBM2, respectively. DAE2 layer is the output layer of the denoising module, and also the input layer of the two-layer RBM module. RBM2 is the output layer of the DDNN model which represents the feature of the document, and it will be compared with the visual layer **v**. This layer is the high-level feature representation of the text data. The subsequent text classification task is also addressed on the basis of this vector. For all nodes, there is no connection between the same layer nodes, but the nodes between those two layers are fully connected.

Specifically, the introduction of the energy model is to capture the correlation between variables, while optimizing the model parameters. Therefore, it is important to embed the optimal solution problem into the energy function when training the model parameters. Here, RBM energy function is defined as

$$E(v,h) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i. \quad (1)$$

Here, (1) represents the energy function of each visible node and hidden node connection structure. Among them, n is the number of hidden nodes, m is the number of visible layer nodes, and b and c are the bias of visual layer and hidden layer, respectively. The objective function of the RBM model is to accumulate the energy of all the visible nodes and the hidden nodes. Therefore, it is necessary for each sample to count the value of all the hidden nodes corresponding to it, so that the total energy can be calculated. The calculation is complex. An effective solution is to convert the problem into probabilistic computing. The joint probability of the visible and the hidden node is

$$P(v,h) = \frac{e^{-E(v,h)}}{\sum_{v,h} e^{-E(v,h)}}. \quad (2)$$

By introducing this probability, the energy function can be simplified, and the objective of the solution is to minimize the energy value. There is a theory in statistical learning that the state of low energy has higher probability than high energy, so we maximize this probability and introduce the

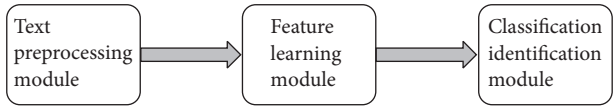


Figure 6: The architecture of a classifier.

free energy function. The definition of free energy function is as follows:

$$\text{FreeEnergy}(v) = -\ln \sum_{h} e^{-E(v,h)}. \tag{3}$$

Therefore,

$$P(v) = \frac{e^{\text{FreeEnergy}(v)}}{Z}, \quad Z = \sum_{v,h} e^{-E(v,h)}, \tag{4}$$

where Z is the normalization factor. Then, the joint probability $P(v)$ can be transformed into

$$\ln P(v) = -\text{FreeEnergy}(v) - \ln Z. \tag{5}$$

The first term on the right side of (5) is the negative value of the sum of the free energy functions of the whole network, and the left is the likelihood function. As we described in the model description, the model parameters can be solved using maximum likelihood function estimation.

Here, we first construct a denoising function module for the original features. It is mainly composed of a DAE. The two-layer DAE is placed at the bottom of the model so as to make full use of the character of denoising. The input signal can be denoised by reconstructing the input signal through unsupervised learning, so that the signal entering the network is purer after being processed by the encoder. Then the impact of noise data on the subsequent construction of the classifier will be reduced.

The second module is developed using DBN. It is generated through RBM; then the ability of feature extraction in this model will be improved. Furthermore, the model can obtain the complex rules in the data, and the high-level features extracted are more representative. In order to achieve better sorting results, we use the extracted representative feature as an input for the final classifier after further extraction using RBM.

Considering the complexity of the training and the efficiency of the model, a two-layer DAE and a two-layer RBM will be used.

3.2. Text Classification Using DDNN. Here, the final DDNN-based text classifier is developed. And there are three key modules in its architecture, as shown in Figure 6.

3.2.1. Text Preprocessing Module. First, the feature words processed here are mapped into the vocabulary form [31–33]. Then, the weights are counted using TF-IDF (term frequency, inverse document frequency) algorithm [34]. In addition, using vector to represent the text is implemented. Meanwhile, it is also normalized.

3.2.2. Feature Learning Module. The DDNN mentioned in Section 3.1 is used to implement feature learning.

3.2.3. Classification Identification Module. In this module, we use Softmax classifier in classification, and its input is the feature which is learned from the feature learning module. In the classifier, the hypothetical text dataset has n texts from k categories, where the training set is expressed as $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n-1)}, y^{(n-1)}), (x^{(n)}, y^{(n)})\}$ and $x^{(i)}$ represents the ith training text, and y represents different categories ($y^{(i)} \in \{1, 2, \ldots, k-1, k\}$). The main purpose of the algorithm is to calculate the probability of x belonging to the tag category, for the given training set x. Here, that function is as shown in

$$\begin{aligned} h_\theta\left(x^{(i)}\right) &= \begin{bmatrix} P\left(y^{(i)} = 1 \mid x^{(i)}; \theta\right) \\ P\left(y^{(i)} = 2 \mid x^{(i)}; \theta\right) \\ \vdots \\ P\left(y^{(i)} = k \mid x^{(i)}; \theta\right) \end{bmatrix} \\ &= \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^i} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}. \end{aligned} \tag{6}$$

Each subvector of vector $h_\theta(x^{(i)})$ is the probability value that x belongs to different tag categories, and the probability value is required to be normalized, so that the sum of probability value of all the subvectors is 1. And $\theta_1, \theta_2, \ldots, \theta_{k-1}, \theta_k \in \mathbb{R}^{n+1}$ represents the parameter vectors, respectively.

After getting θ, we can obtain the previously assumed function $h_\theta(x)$. It can be used to calculate the probability value that text x belongs to each category. The category which has the biggest probability value is the final classified result by the classifier algorithm.

4. Simulation Results and Discussions

In this article, simulations are conducted in two steps. First, we analyze the key parameters that affect the performance of the DAE and the RBM models (the basic components of DDNN model) and implement the simulation with appropriate parameters. Second, we compare the DDNN with NB, KNN, SVM, and DBN using the data with noise and the data without noise and verify the effectiveness of the proposed DDNN.

4.1. Evaluation Criterion of Text Classification Results. For the text classification results, we mainly use the accuracy as a classification criterion. This index is widely used to evaluate the performance in the field of information retrieval and statistical classification.

If there are two categories of information in the original sample, there are a total of P samples which belong to

category 1, and category 1 is positive. And there are a total of N samples which belong to category 0, and category 0 is negative.

After the classification, TP samples that belong to category 1 are divided into category 1 correctly, and FN samples are divided into category 0 incorrectly. And TN samples that belong to category 0 are divided into category 0 correctly, FP samples are divided into category 1 incorrectly.

Then, the accuracy is defined as

$$\text{Accuracy} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{7}$$

Here, the accuracy can reflect the performance of the classifier.

The recall is defined as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = 1 - \frac{\text{FN}}{P}. \tag{8}$$

It can reflect the proportion of the positive samples classified correctly.

The F-score is defined as

$$F\text{-score} = \frac{2 \times \text{Recall} \times \text{Accuracy}}{\text{Recall} + \text{Accuracy}}. \tag{9}$$

It is a comprehensive reflection of the classification of data.

4.2. Dataset Description. In our simulations, we test the algorithm performance using two news datasets, namely, 20-Newsgroups and BBC news datasets.

The 20-Newsgroups dataset consists of 20 different news comment groups in which each group represents a news topic. There are three versions in the website (http://qwone.com/~jason/20Newsgroups/). And we select the second version, that is, a total of 18846 documents, and the dataset has been divided into two parts, where there are 11314 documents for the train set and 7532 documents for the test set. The distribution of the 20 sample details can be found in that website. Note that, in our simulations, the serial number of those 20 labels varies from 0 to 19.

The dataset of BBC news consists of several news documents on the BBC website (http://www.bbc.co.uk/news/business/market_data/overview/). The dataset includes a total of 2225 documents corresponding to five topics, that is, business, entertainment, politics, sports, and technology. Similarly, we randomly select 1559 documents for train set, and 666 documents for a test set.

4.3. Simulation Results. All the simulations are conducted according to the following. The operating system is Ubuntu 16.04. The hardware environment is NVIDIA Corporation GM204GL [Tesla M60]. The software environment is Cuda V8.0.61 and cuDNN 5.1. Deep learning framework is Keras, while using sklearn and nltk toolkits.

4.3.1. Impact of Parameters. For all deep learning algorithms, the parameter tuning greatly affects the performance of simulation results. For the DDNN, the parameters which we mainly adjust include the plus noise ratio of the data, the number of hidden layer nodes, and the learning rate.

In order to test the robustness of the DDNN, we set the plus noise ratio of the training set to 0.01, 0.001, and 0.0001. The result are shown in the Table 1.

As shown in Table 1, the stability of the model can be guaranteed within the range of plus noise ratio (0.01, 0.001), but when the plus noise ratio is too high, that is, higher than 0.1, the data will be damaged especially for the sparse data, and it will affect the classification performance. Moreover, the performance of the classifier to robust feature extraction will be weakened if the plus noise ratio is too low. Hence, we set the plus noise ratio finally to 0.001. After we conduct the simulation, we set the noise factor as 0.01, 0.02, 0.03, 0.04, and 0.05 to verify the denoising performance of the proposed model.

The number of the input layer nodes is fixed according to the result of the weight using TF-IDF algorithm. Since the main purpose of DAE is to reconstruct original data, we set the numbers of the input layer nodes and output layer nodes to the same value. Because the number of the hidden layer nodes is unknown, we set the numbers of the two hidden-layer nodes in DAE to 1600 and 1500, 1700 and 1500, and 1800 and 1500, respectively. In addition, the numbers of the two hidden-layer nodes in RBM are set as 600 and 100, 700 and 100, and 800 and 100, respectively. Then, we conduct the simulation. And we set the learning rate to 0.1, 0.01, and 0.001. The results are shown in Table 2.

As shown in Table 2, the performance of the DDNN model will be better when the numbers of two hidden-layer nodes are set to 1700 and 1500 for DAE and 700 and 100 for RBM, respectively. And the learning rate should be set to 0.01.

4.3.2. Comparisons and Analysis. In this article, we compare our DDNN model with NB, KNN, SVM, and DBN models.

In text preprocessing, we select the frequency of the first 2000 words to simulation and set batch size with 350. Compared with the DDNN model (two-layer DAE and two-layer RBM) proposed in this article, the DBN model is also set to four layers. The number of iterations in the pretraining phase is 100, and the model updating parameter is 0.01.

Here, we take the BBC news dataset for an example to show the process of training. From Figures 7 and 8, we can see that, with the increase of epoch, the loss of training is decreasing and the accuracy is increasing towards test datasets, which shows that the effect of training is well.

Table 3 compares the results of DDNN with other models using the BBC news dataset and Table 4 compares them using the 20-Newsgroups dataset. Moreover, we compare these models in consideration of different types of data, including the data without noise and the data with a noise factor of 0.01, 0.02, 0.03, 0.04, and 0.05. Here, it is noted that, for each vector of text extracted, the standard normal distribution of noise factor multiplication is added. If a dimension is less than 0, it is directly set to 0. In this article, the accuracy rate (Accuracy), recall rate (Recall), and F-Score are observed to evaluate the performance of classifier. Take the calculation of Accuracy, for example. Towards each classifier, we firstly calculate the accuracy of each category according to the metric (7) and

Table 1: Text classification performance of DDNN with different plus noise ratio.

Plus noise ratio	Noise factor					
	0.00	0.01	0.02	0.03	0.04	0.05
0.001	0.7530	0.7529	0.7479	0.7450	0.7349	0.7287
0.01	0.7536	0.7561	0.7550	0.7542	0.7443	0.7378
0.1	0.5379	0.5310	0.5270	0.5179	0.5027	0.4978

Table 2: Text classification performance of DDNN with different parameters.

Learning rate	DAE		RBM		Accuracy
0.01	1600	1500	600	100	0.9640
	1700	1500	700	100	0.9700
	1800	1500	800	100	0.9686
0.02	1600	1500	600	100	0.9655
	1700	1500	700	100	0.9654
	1800	1500	800	100	0.9670
0.03	1600	1500	600	100	0.9625
	1700	1500	700	100	0.9627
	1800	1500	800	100	0.9491

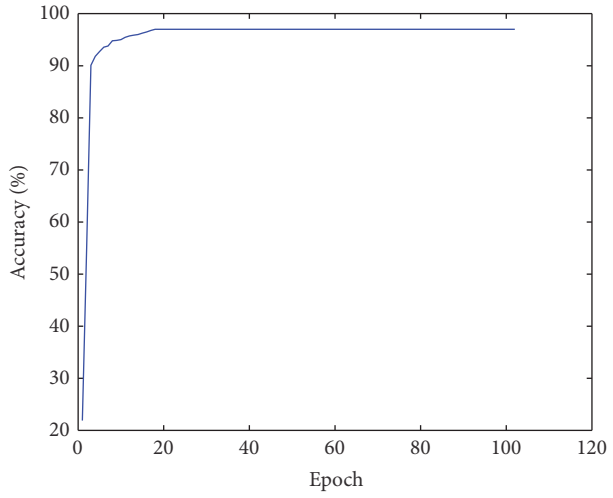


Figure 7: The test accuracy in the training process for BBC news dataset.

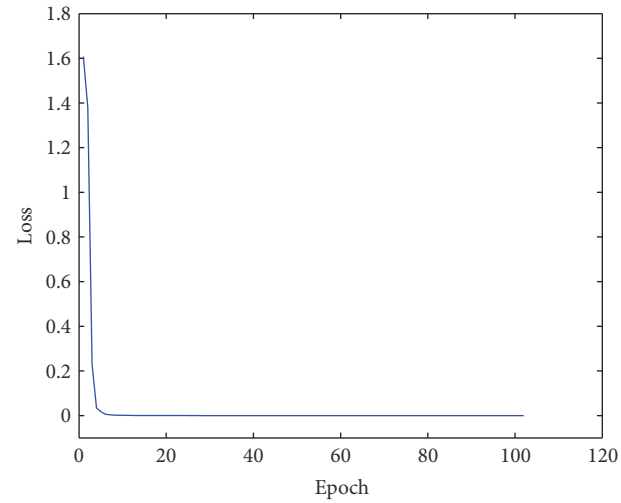


Figure 8: The test loss in the training process for BBC news dataset.

then compute the average of these subaccuracies as the result. The simulation data is the optimal classification result after running many times.

After comparing DDNN model with shallow submodel, including KNN and SVM, from those analysis results in Tables 3 and 4, DDNN achieves a better performance. The reason is that when the training set is sufficient, the DDNN can be fully trained, so that the parameters of the network itself can reach the optimal value as much as possible to fit the distribution of training data, and the high-level features extracted from the underlying features are more discriminative for the final classification function.

Compared with the DBN model, DDNN first uses the DAE model to train the classification results more accurately in the case that the two layers of the model are the same (they are all four layers). This is because the first two layers in the DDNN model are with DAE, which can effectively reduce the impact of noise data, and the DDNN model can be more flexible to adjust the parameters. On the other hand, due to the use of DAE as the initial layer, the dimension of data can also be reduced preliminary.

As shown in Tables 3 and 4, the classification performance of NB, KNN, and SVM is obviously decreased when the dataset is adjusted with noise factor, and the DNNN has better antinoise effect for only about 1% decline.

Furthermore, Table 5 shows the running time of different models. We can easily find that, for each sample, the NB classifier holds the shortest running time and SVM classifier holds the longest running time. Meanwhile, it can be seen that

Table 3: Text classification performance with different models using BBC news dataset.

	Classifier	Plus ratio noise					
		0.00	0.01	0.02	0.03	0.04	0.05
Accuracy	NB	0.9659	0.9560	0.9339	0.8736	0.8186	0.7852
	KNN	0.9375	0.9325	0.9284	0.9373	0.9119	0.9260
	SVM	0.9715	0.9701	0.9672	0.9583	0.9340	0.9075
	DBN	0.9462	0.9434	0.9268	0.9076	0.8789	0.8479
	DDNN	*0.9700*	*0.9685*	*0.9582*	*0.9541*	*0.9381*	*0.9286*
Recall	NB	0.9655	0.9550	0.9294	0.8453	0.7387	0.6652
	KNN	0.9354	0.9324	0.9279	0.9369	0.9114	0.9249
	SVM	0.9715	0.9700	0.9670	0.9580	0.9309	0.8964
	DBN	0.9459	0.9429	0.9249	0.9039	0.8769	0.8393
	DDNN	*0.9700*	*0.9685*	*0.9580*	*0.9535*	*0.9399*	*0.9249*
F-score	NB	0.9657	0.9555	0.9316	0.8592	0.7766	0.7202
	KNN	0.9364	0.9324	0.9281	0.9371	0.9116	0.9254
	SVM	0.9715	0.9700	0.9671	0.9581	0.9324	0.9019
	DBN	0.9460	0.9431	0.9258	0.9057	0.8779	0.8436
	DDNN	*0.9700*	*0.9685*	*0.9581*	*0.9538*	*0.9390*	*0.9267*

Table 4: Text classification performance with different models using 20-Newsgroup dataset.

	Classifier	Noise factor					
		0.00	0.01	0.02	0.03	0.04	0.05
Accuracy	NB	0.7506	0.7274	0.6895	0.6678	0.5887	0.4633
	KNN	0.6136	0.6161	0.6213	0.6142	0.6043	0.5978
	SVM	0.7598	0.7527	0.7294	0.6968	0.6652	0.6453
	DBN	0.7235	0.7207	0.7041	0.6849	0.6562	0.6252
	DDNN	*0.7536*	*0.7561*	*0.7550*	*0.7542*	*0.7443*	*0.7378*
Recall	NB	0.7483	0.6693	0.5053	0.3526	0.2613	0.2027
	KNN	0.5959	0.6000	0.6070	0.6034	0.5939	0.5820
	SVM	0.7525	0.7415	0.6966	0.6094	0.4891	0.3833
	DBN	0.7149	0.7120	0.6990	0.6826	0.6439	0.6250
	DDNN	*0.7459*	*0.7500*	*0.7549*	*0.7534*	*0.7439*	*0.7320*
F-score	NB	0.7494	0.6971	0.5832	0.4615	0.3619	0.2820
	KNN	0.6046	0.6079	0.6141	0.6088	0.5991	0.5898
	SVM	0.7561	0.7471	0.7126	0.6502	0.5637	0.4809
	DBN	0.7192	0.7163	0.7015	0.6837	0.6500	0.6251
	DDNN	*0.7497*	*0.7530*	*0.7549*	*0.7538*	*0.7441*	*0.7349*

Table 5: The running time of different models (ms).

Classifier	Dataset	
	BBC news	20-Newsgroups
NB	0.005	0.006
KNN	0.150	0.870
SVM	1.660	12.060
DBN	0.110	0.180
DDNN	*0.120*	*0.210*

the DDNN classifier can keep good classification speed while achieving good classification performance.

5. Conclusion

This article combines the DAE and RBM to design a novel DNN model, named DDNN. The model first denoises the data based on the DAE and then extracts feature of the text effectively based on RBM. Specifically, we conduct the simulations on the 20-Newsgroups and BBC news datasets and compare the proposed model with other traditional classification algorithms, for example, NB, KNN, SVM, and DBN models, considering the impact of noise. It is verified that the DDNN proposed in this article achieves better antinoise performance, which can extract more robust and deeper features while improving the classification performance.

Although the proposed model DDNN has achieved satisfactory performance in text classification, the text used in

the simulations is long-type data. However, considering that there are also some short text data in text classification task, we should address this issue using the model DDNN. Moreover, to further improve the computational performance in the implementation of deep learning methods, in the future we can also design some hybrid learning algorithms by incorporating some advanced optimization techniques, for example, kernel learning and reinforcement learning, into the framework of DDNN, while applying it in some other fields.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research is funded by the Fundamental Research Funds for the China Central Universities of USTB under Grant FRF-BD-16-005A, the National Natural Science Foundation of China under Grant 61174103, the National Key Research and Development Program of China under Grants 2017YFB1002304 and 2017YFB0702300, the Key Laboratory of Geological Information Technology of Ministry of Land and Resources under Grant 2017320, and the University of Science and Technology Beijing-National Taipei University of Technology Joint Research Program under Grant TW201705.

References

[1] A. M. Rinaldi, "A content-based approach for document representation and retrieval," in *Proceedings of the 8th ACM Symposium on Document Engineering (DocEng '08)*, pp. 106–109, ACM, São Paulo, Brazil, September 2008.

[2] E. Baykan, M. Henzinger, L. Marian, and I. Weber, "A comprehensive study of features and algorithms for URL-based topic classification," *ACM Transactions on the Web*, vol. 5, no. 3, article 15, 2011.

[3] P. Langley, W. Iba, and K. Thompson, "An analysis of bayesian classifiers," in *Proceedings of the 10th National Conference on Artificial Intelligence*, pp. 223–228, San Jose, Calif, USA, 1992.

[4] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *Proceedings of the 15th National Conference on Artificial Intelligence—Workshop on Learning for Text Categorization*, pp. 41–48, Madison, Wis, USA, 1998.

[5] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *Proceedings of the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*, pp. 42–49, Berkeley, Calif, USA, August 1999.

[6] S. Godbole, S. Sarawagi, and S. Chakrabarti, "Scaling multi-class support vector machines using inter-class confusion," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 513–518, Edmonton, Canada, July 2002.

[7] S. L. Y. Lam and D. L. Lee, "Feature reduction for neural network based text categorization," in *Proceedings of the 6th International Conference on Database Systems for Advanced Applications*, pp. 195–202, Hsinchu, Taiwan, 1999.

[8] M. E. Ruiz and P. Srinivasan, "Hierarchical neural networks for text categorization," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 281-282, Berkeley, Calif, USA, August 1999.

[9] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, article 1883, 2009.

[10] X. Luo, J. Deng, J. Liu, W. Wang, X. Ban, and J. Wang, "A quantized kernel least mean square scheme with entropy-guided learning for intelligent data analysis," *China Communications*, vol. 14, no. 7, pp. 127–136, 2017.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[12] D. Silver, A. Huang, C. J. Maddison et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[13] G. E. Hinton, "Deep belief networks," *Scholarpedia*, vol. 4, no. 5, article 5947, 2009.

[14] P. Smolensky, "Information processing in dynamical systems: foundations of harmony theory," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, D. E. Rumelhart and J. L. McLelland, Eds., pp. 194–281, MIT Press, 1986.

[15] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F. F. Li, "ImageNet: a large-scale hierarchical image database," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 248–255, Miami, Fla, USA, June 2009.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[17] P. Vincent, H. Larochelle, and Y. Bengio, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103, ACM, Helsinki, Finland, July 2008.

[18] P. Vincent, H. Larochelle, and I. Lajoie, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[19] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[20] X. Luo, Y. Xu, W. Wang et al., "Towards enhancing stacked extreme learning machine with sparse autoencoder by correntropy," *Journal of the Franklin Institute*, 2017.

[21] R. Collobert, J. Weston, and L. Bottou, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[22] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning—a new frontier in artificial intelligence research," *IEEE Computational Intelligence Magazine*, vol. 5, no. 4, pp. 13–18, 2010.

[23] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[24] X. Wei and W. B. Croft, "LDA-based document models for ad-hoc retrieval," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 178–185, Seattle, Wash, USA, August 2006.

[25] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder," in *Proceedings of the 14th*

Annual Conference of the International Speech Communication Association, pp. 436–440, Lyon, France, August 2013.

[26] N. Le Roux and Y. Bengio, "Representational power of restricted Boltzmann machines and deep belief networks," *Neural Computation*, vol. 20, no. 6, pp. 1631–1649, 2008.

[27] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–27, 2009.

[28] A. Fischer and C. Igel, "An introduction to restricted Boltzmann machines," in *Proceedings of the 17th Iberoamerican Congress on Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 14–36, Buenos Aires, Argentina, 2012.

[29] L. F. Polana and K. E. Barner, "Exploiting restricted Boltzmann machines and deep belief networks in compressed sensing," *IEEE Transactions on Signal Processing*, vol. 65, no. 17, pp. 4538–4550, 2017.

[30] R. Karakida, M. Okada, and S.-I. Amari, "Dynamical analysis of contrastive divergence learning: Restricted Boltzmann machines with Gaussian visible units," *Neural Networks*, vol. 79, pp. 78–87, 2016.

[31] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 3111–3119, Lake Tahoe, Calif, USA, 2013.

[32] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 28th Annual Conference on Neural Information Processing Systems*, pp. 3104–3112, Montreal, Canada, 2014.

[33] M. Zhong, H. Liu, and L. Liu, "Method of semantic relevance relation measurement between words," *Journal of Chinese Information Processing*, vol. 23, no. 2, pp. 115–122, 2009.

[34] L. P. Jing, H. K. Huang, and H. B. Shi, "Improved feature selection approach TFIDF in text mining," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 944–946, Beijing, China, 2002.

Hindawi
Scientific Programming
Volume 2017, Article ID 2938369, 8 pages
https://doi.org/10.1155/2017/2938369

Research Article

Advertisement Click-Through Rate Prediction Based on the Weighted-ELM and Adaboost Algorithm

Sen Zhang,[1,2] Qiang Fu,[1,2] and Wendong Xiao[1,2]

[1]*School of Automation & Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China*
[2]*Key Laboratory of Knowledge Automation for Industrial Processes, Ministry of Education, Beijing 100083, China*

Correspondence should be addressed to Sen Zhang; zhangsen@ustb.edu.cn

Received 13 July 2017; Accepted 4 October 2017; Published 9 November 2017

Academic Editor: Wenbing Zhao

Accurate click-through rate (CTR) prediction can not only improve the advertisement company's reputation and revenue, but also help the advertisers to optimize the advertising performance. There are two main unsolved problems of the CTR prediction: low prediction accuracy due to the imbalanced distribution of the advertising data and the lack of the real-time advertisement bidding implementation. In this paper, we will develop a novel online CTR prediction approach by incorporating the real-time bidding (RTB) advertising by the following strategies: user profile system is constructed from the historical data of the RTB advertising to describe the user features, the historical CTR features, the ID features, and the other numerical features. A novel CTR prediction approach is presented to address the imbalanced learning sample distribution by integrating the Weighted-ELM (WELM) and the Adaboost algorithm. Compared to the commonly used algorithms, the proposed approach can improve the CTR significantly.

1. Introduction

With the development of the network technology and the communication technology, the Internet and the mobile Internet have been developed rapidly. Due to the popularity of smart phones, a variety of the mobile phone applications are invented. It is a niche market where the advertisers and the advertising companies pay more attention to the click-through rate (CTR) in the online advertising products. Usually the online advertising can be done in two different ways: one is the website search based advertising, which specifically refers to the searching engine depending on the user's key words that target the advertising content and the advertising spot. The other one is the real-time bidding (RTB) advertising, in which the advertising supplier platform provides no longer the advertising spot, but the specific users who visited the advertisement spot. The RTB advertisements enlarge the online advertising's directivity and accuracy [1].

Currently, there exists many research works on CTR prediction for Internet advertising. Menon et al. [2] proposed the maximum likelihood algorithm to estimate the parameters of the CTR probabilistic model. But this model can only be applied to the existing advertisements rather than the new advertisements. Richardson et al. [3] proposed the logic regression model to learn the CTR prediction model for searching advertising with the model features including the number of the keywords, the position of the figures in the page, and the other characteristics of the advertisements. Chapelle [4] proposed a stochastic regression approach based on the rate estimation machine learning framework for the Yahoo! to solve the CTR prediction problem by using four features as the model inputs. The norm-2 regularization term is added in the logistic regression model. This method can produce a sparser model to increase the number of the nonzero parameters to avoid the overfitting problem. Shao [5] proposed a high-level feature representation and a click-by-point prediction method based on the deep network that combines the high-level features and the basic features by using deep neural network model.

Most existing work on CTR prediction is focused on searching advertising that is seriously dependent on the keyword and the user input. With the development of the intelligent terminals and the mobile Internet, RTB advertising is increasing rapidly. More and more advertisers are in favor of the RTB advertising which will become the main trend of the Internet advertising in the future. At the same time, the

Table 1: Description of the experimental dataset.

Attribute name	Data type	Attribute explanation
push_time	Timestamp	Time of bidding request
u_id	String	User ID
exchange_id	Int	Advertising Exchange Platform ID
c_id	String	Advertising Creative ID
space_id	String	Advertising Position ID
area_id	Int	Area ID
media_id	Int	Meida ID
advertiser_id	Int	Advertiser ID
policy_id	Int	Policy ID
user_agent	String	Agent of browser
user_ip	String	User ip
if_click	Int	If click
if_show	Int	If show
price_base	Double	Lowest price for bidding the advertising position
price_win	Double	Price for winning the advertising position
url	String	URL

research work on the RTB CTR prediction is still at the beginning stage.

In this paper, we will study the novel big data based online CTR prediction problem by incorporating RTB advertising with user profile system. A novel CTR prediction approach will be presented by integrating the Weighted-ELM (WELM) and the Adaboost algorithm to address the imbalanced learning sample distribution. We will perform the experiments using real advertising datasets to verify the effectiveness of the proposed approach.

2. The Experimental Dataset and the Evaluation Criteria

In this section, the experimental dataset and the evaluation criteria used in this study named Area Under Curve (AUC) will be briefly described.

The experimental dataset used in this paper for CTR prediction is the original data log provided by a domestic advertising company in China. There are 16 attributes in the original data log, with the details shown in Table 1.

2.1. User Profile. Since the advertising log has large amount of data, we divide the above 16 attributes into 4 categories: the user's characteristics, the temporal characteristics, the ID characteristics, and the numerical characteristics.

2.1.1. The User's Characteristics. In early practice, when the demand side platform receives the bidding request from the advertising agent, normally the user's information is not analyzed and all users were used for advertising. It is proved that this way of the information delivery cannot achieve the desired results as the u_id and media_id attributes used in the approach cannot cater to the users' interest. Thus the primary task is to establish the user profile system to obtain the user's age, gender, and interest preference for CTR prediction. The overall structure of the system is shown in Figure 1.

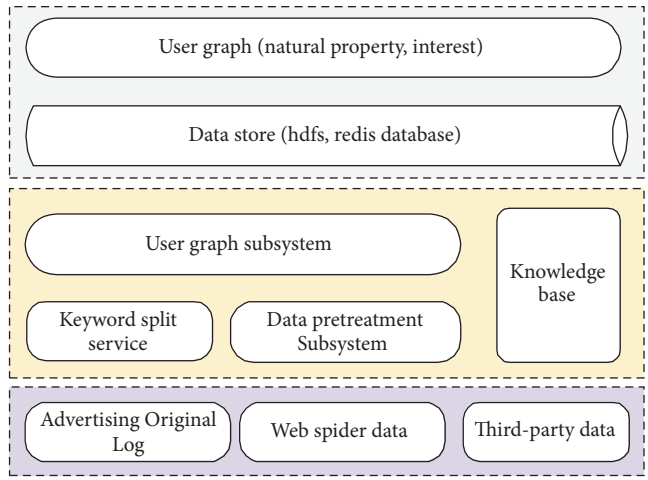


Figure 1: The block diagram of the user profile system.

The user profile system mainly includes the following functions:

(i) Data pretreatment subsystem: take the responsibility of cleaning and preprocessing the advertising log data;

(ii) Keyword split service: take the responsibility of segmenting the irregular text;

(iii) Knowledge base: take the responsibility of providing the related mapping tables;

(iv) User graph subsystem: the most important part of the user graph system: take the responsibility of integrating various parts of the data to build a user graph;

(v) Data storage subsystem: take the responsibility of storing the results of the user graph.

The output of the user graph system includes the user's age, gender, and interest preference. The users' characteristics are obtained by using i_id attribute to match the output of the user graph system.

2.1.2. The Time Characteristics. The time characteristics include the field of push_time in the log which represents the time of the ads request. According to the historical data, the users have different interests at different time periods, so the probability of a click behavior is also different. Based on this judgment, we split one day into six time periods which are late-night, morning, lunch time, afternoon, dinner time, and evening. The entire time information is organized by a six-dimensional vector. The six periods of time are shown in Table 2.

2.1.3. The ID Characteristics. The ID characteristics in the dataset include the u_id, the advertiser_id, the media_id, the

Table 2: Information of a whole day.

Period name	Period
T1 (late-night)	00:00~06:00
T2 (morning)	06:00~11:00
T3 (lunchtime)	11:00~13:00
T4 (afternoon)	13:00~18:00
T5 (dinnertime)	18:00~20:00
T6 (evening)	20:00~23:00

area_id, the c_id, the policy_id, and the exchange_id. There are a lot of ID attributes in the RTB advertising logs. If we do not have the filtering process of the characteristics, we would obtain a vector whose dimension may be up to several hundred thousands which increases the computational complexity seriously. Therefore, it is necessary to reduce the dimensionality of the feature space. We apply the method in [3] to remove the needless ID attributes that have no impact or little impact on the click-through rate.

2.1.4. The Numerical Characteristics. Attributes in the dataset, such as the price_base, the price_win, the URL, and the u_ip, affect the advertising's CTR as well. Take the price_win for example, if the value is 0, it indicates that the advertising is not a successful bidding. If the value is nonzero, the different values reflect that the value of the advertising clicking is different. It is usually considered that the larger the value is, the better the advertising position is and the greater the probability of the clicking is. Therefore the numerical attributes need to be added to the feature vector.

In this paper, we adopted the maximum and minimum normalization method to normalize each characteristic to the value between 0 and 1.

2.2. Area Under Curve (AUC). The prediction of the CTR is a binary classification problem while the proportion of the positive and negative samples is extremely uneven. In the actual advertising, the proportion of the positive and the negative samples is about 3 : 1000 or even lower. The samples are distributed in different categories unevenly, so the evaluation index of accuracy is not a good criterion to judge the performance of the classifier.

In this paper, AUC is adopted to measure the effect of the CTR prediction. In the process of calculating the AUC, the related curve is called ROC curve (receiver operating characteristics) [6]. Traditional ROC curve is used in medical field. Currently it is often used in the field of data mining, machine learning, and pattern recognition.

When the ROC curve is drawn, the horizontal coordinate is FPR (False Positive Rate) and the vertical coordinate is TPR (True Positive Rate). The values of FPR and TPR can be calculated according to the formula (1).

$$\begin{aligned} \mathrm{FPR} &= \frac{\mathrm{FP}}{\mathrm{N}} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \\ \mathrm{TPR} &= \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \end{aligned} \tag{1}$$

Table 3: The AUC value of each ID attribute.

ID attribute	exchange_id	area_id	media_id	advertiser_id
AUC value	0.5137	0.5412	0.5930	0.5271

In (1), TP represents the fact that the samples are positive and the algorithm recognizes them as the positive samples; FP represents the fact that the samples are negative and the algorithm recognizes them as the positive samples; FN represents the fact that the samples are positive and the algorithm recognizes them as the negative samples; TN represents the fact that the samples are negative and the algorithm recognizes them as the negative samples [7].

It is obvious that if there are more users to click an advertisement, the rank of this advertisement will be in the front and the area under the ROC curve is larger which indicates that the performance of the advertising is better.

As an example, we draw the receiver operating characteristics (ROC) curves for the exchange_id, the area_id, the media_id, and the advertiser_id by the Weighted-ELM. Each AUC value of the curve is shown in Table 3.

From Table 3, we can see that the AUC values of the exchange_id and the advertiser_id are almost 0.5, which have no difference from the random results. This phenomenon has something related to the characteristics of the RTB advertising. The RTB advertisers do not want their own click conversion data to be used to optimize the other advertisers' effectiveness.

Compared to the AUC value of the advertiser_id, the AUC value of the media_id is increased slightly and up to 0.60. This case is related to the user's interest and the media_id can reflect the user's interest. If the users visit a few apps frequently, the probability of clicking the ads would be increased.

3. The CTR Assessment

In this section, the ELM algorithm will be discussed, which will be used in the prediction of the CTR. Compared with the traditional classification algorithms SVM and BP, the ELM has the advantage of fast learning speed and accurate estimation results with easily setting the weights. Based on these advantages, the ELM algorithm has been developed rapidly since it was proposed several years ago. Because the proportion of the positive and the negative samples is extremely uneven, we proposed the Weighted-ELM algorithm to solve the problem in the next subsection. Because the ELM is the basis of the Weighted-ELM algorithm, we will firstly describe the original ELM in the following.

3.1. The ELM Algorithm. In recent years, Huang et al. [8–10] and the other scholars proposed a fast algorithm of single-hidden layer feedforward neural network named extreme learning machine (ELM) [11, 12]. The specific structure of the ELM algorithm is shown in Figure 2.

The input weights and the bias of the hidden node in the ELM are chosen randomly. They do not need a series of iterative algorithm, which greatly saves the training time of

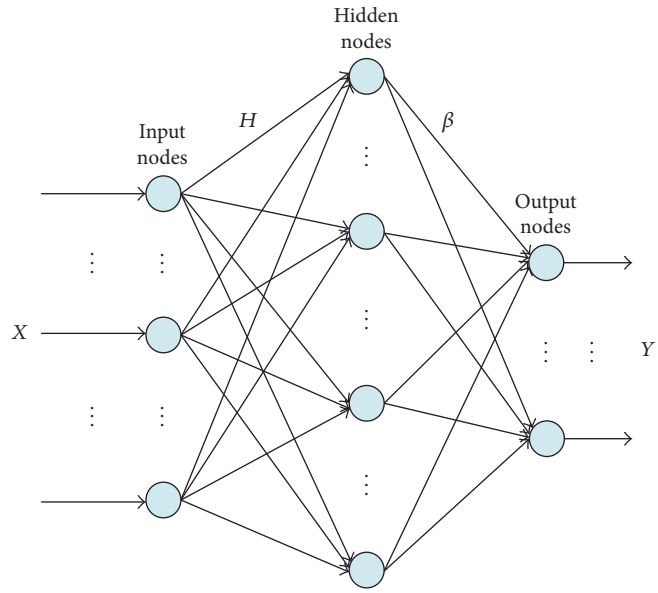


Figure 2: The structure of the ELM neural network.

the neural network. The output weights of the ELM are obtained by minimizing the squared error loss function to get the least square solution. Thus the process of determining the neural network parameters is very simple that saves much time of adjusting the parameters.

The basic idea of the ELM algorithm is as follows.

The training sample set is given as $\{(X, T) \mid X = [X_1, X_2, \ldots, X_N]^T,\ T = [t_1, t_2, \ldots, t_N]\}$, where the matrix X is the input matrix of the neural network and the matrix T is the actual output value of the training sample set. From the neural network with L hidden nodes, we can get

$$\sum_{i=1}^{L} \beta_i \cdot G\left(a_i \cdot X_j + b_i\right) = O_i, \quad j = 1, 2, \ldots, N. \tag{2}$$

In this equation, $G(aX + b)$ is the neural network hidden layer node activation function. Usually it is sig, sin, hardlim, or tribas function; a_i is the connection weights between the ith hidden layer node and the input nodes; b_i is the bias of the ith hidden node; β_i is the connection weights between the ith hidden layer nodes and the output node.

In the practical application of the algorithm, the output value of the network is equal or near to the actual output value. If the sample set and the neural network structure are close to the target value T with the zero error, we can get $\|H\beta - T\| = 0$. The formula of the ELM algorithm can be abbreviated as

$$H\beta = T, \tag{3}$$

where H is the output matrix of the neural network hidden nodes and β is the output weight matrix between the hidden layer nodes and the output layer node.

The main idea of the algorithm is how to get the output weight matrix β to make the training error $\|H\beta - T\|^2$ and the output weight matrix $\|\beta\|$ minimum. That means how to make the following equation's value minimum:

$$\begin{aligned} J &= (H\beta - T)^T (H\beta - T) \\ \beta &= H^{\Gamma} T, \end{aligned} \tag{4}$$

where H^{Γ} is the generalized inverse matrix of H. If $H^T H$ is nonsingular, $H^{\Gamma} = (H^T H)^{-1} H^T$. If HH^T is nonsingular, $H^{\Gamma} = H^T (HH^T)^{-1}$. If H is not full column rank, β could be obtained by the singular value decomposition (SVD) [5, 13].

3.2. The Weighted-ELM Algorithm. The basic ELM algorithm is very useful for many problems. However, there exist a lot of classification problems whose samples are imbalance, such as the advertising click rate problem. In order to solve the problem of the sample imbalance in the classification, Xu et al. proposed the Weighted-ELM algorithm [14].

The objective function of the ELM algorithm is

$$L_{\min} = \frac{1}{2} \|\beta\|^2 + \frac{C}{2} \sum_{i=1}^{N} \xi_i^2. \tag{5}$$

In this equation, the condition is satisfied: $h(x_i)\beta = t_i - \xi_i, i = 1, 2, \ldots, N$. The first half of formula (5) is called the structural risk, and the latter part is called the empirical risk.

The objective function of the Weighted-ELM algorithm is

$$L_{\min} = \frac{1}{2} \|\beta\|^2 + \frac{C}{2} \mathbf{W} \sum_{i=1}^{N} \xi_i^2, \tag{6}$$

where $\mathbf{W}$ is an $\mathbf{N} \times \mathbf{N}$ diagonal matrix and the value of the matrix $\mathbf{W}$ is related to each training sample. Generally, if $\mathbf{x}_i$ belongs to a few classes, the corresponding $\mathbf{W}_{ii}$ should be

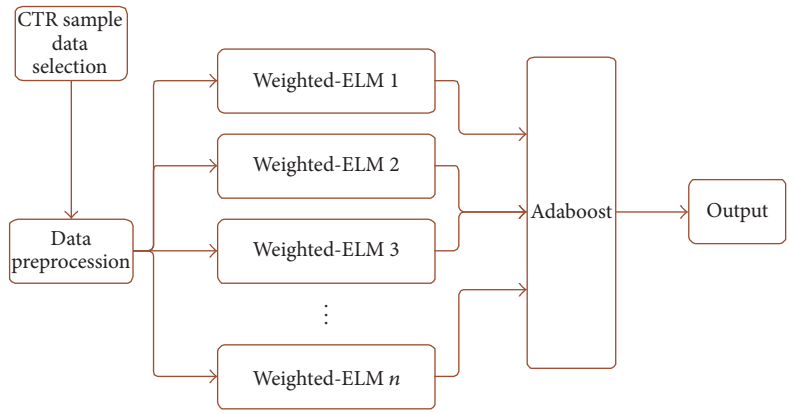


Figure 3: The flowchart of the WELM-Adaboost algorithm.

given a relatively large weight. There are two methods for the value of w. The first method is shown in

$$W_{ii} = \frac{1}{\#(t_i)} \tag{7}$$

$$W_{ii} = \begin{cases} \dfrac{k}{\#(t_i)} & \text{if } t_i > \text{avg}(t_i) \\ \dfrac{1}{\#(t_i)} & \text{if } t_i \le \text{avg}(t_i); \end{cases} \tag{8}$$

the second method is as follows.

The process of training ELM is equivalent to solving the following problem:

$$L_{\text{ELM}} = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}C\mathbf{W}\sum_{i=1}^{N}\xi_i^2 - \sum_{i=1}^{N}\alpha_i\left(\mathbf{h}(\mathbf{x}_i)\beta_i - t_i + \xi_i\right). \tag{9}$$

Similar to the original ELM, $\boldsymbol{\beta}$ is also solved in two ways:

When N is small,

$$\boldsymbol{\beta} = \mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \mathbf{W}\mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{W}\mathbf{T}. \tag{10}$$

When N is large,

$$\boldsymbol{\beta} = \left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{W}\mathbf{T}. \tag{11}$$

The output of the Weighted-ELM classifier can be given by

$$f(x) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta}. \tag{12}$$

4. WELM-Adaboost Algorithm

This paper constructs the advertisement click rate prediction model by the proposed WELM-Adaboost algorithm which can adjust the weight of the data distribution.

4.1. Adaboost Algorithm. Adaboost algorithm is one of the typical applications of the Boosting algorithm. The Adaboost algorithm chooses the very important features to construct a series of weak classifiers and cascade these weak classifiers to form a stronger classifier. The advantage of this algorithm is that it uses the weighted training data instead of the randomly selected training samples. It combines the weak classifiers and uses the weighted voting mechanisms instead of the average voting mechanism.

4.2. The Advertisement Click Rate Prediction Model Based on the WELM-Adaboost. In this paper, the Weighted-ELM is used as a weak predictor, and the weight distribution of each sample is adjusted by using the Adaboost algorithm to obtain multiple Weighted-ELM classifiers. These classifiers are combined into a strong classifier [14].

The advertisement click rate prediction process based on the WELM-Adaboost algorithm is shown in Figure 3.

The detailed steps of the algorithm are as follows:

(1) From the sample data, randomly select N sample data as the training data. According to the positive and the negative samples of the distribution ratio, initialize the weights of each training sample.

(2) For each iteration $m = 1 : M$, where M is the total number of the weak classifiers, the algorithm will repeat the following steps from (a) to (e):

(a) Apply the training samples to a classifier $\text{ELM}_m(x)$ with the initial sample weight w_i;

(b) Calculate the weight prediction error from the weights of the $\text{ELM}_m(x)$ whose results are misclassified samples; the weight prediction error is calculated according to

$$\text{err}_m = \frac{\sum_{i=1}^{N} w_i I\left(y_i \neq \text{ELM}_m(x_i)\right)}{\sum_{i=1}^{N} w_i}. \tag{13}$$

(c) Calculate the weight of the sequence α_m of the $\text{ELM}_m(x)$ according to its classification performance:

Table 4: The statistics of the experimental dataset.

Dataset	Impression_num	Click_num	Click/impression
Training set	180823	544	0.0030
Test set	77496	252	0.0032
total	258319	796	0.0031

$$\alpha_m = \frac{1}{2}\ln\left(\frac{1-\mathrm{err}_m}{\mathrm{err}_m}\right). \quad (14)$$

(d) The weight of the new training sample is adjusted according to the calculated sequence weight α_m:

$$w_i = w_i \cdot \exp\left(\alpha_m \cdot I\left(y_i \neq \mathrm{ELM}_m\left(x_i\right)\right)\right). \quad (15)$$

(e) Renormalize the sample weight.

(3) After M iterations, the M-group weak predictors are obtained. These weak predictors are merged into the final strong predictor $C(x)$:

$$C\left(x\right) = \arg\max_k \sum_{m=1}^{M} \alpha_m \cdot I\left(\mathrm{ELM}_m\left(x\right) = k\right), \quad (16)$$

where k is the number of the categories of the samples.

5. The Experimental Results

The experimental dataset used in this paper is the RTB advertisement raw log data provided by a domestic advertisement company in Beijing, China. Since the data is too large and the positive (or the negative) samples are seriously imbalanced, we randomly extract 1‰ of the data as the experimental data from the log. Click samples are recorded as positive; the other (nonclick) samples are negative. The proportion of the positive and the negative samples of the experimental data is almost 3 : 1000 which is a typical unbalanced data set. The statistics of the experimental data is shown in Table 4. In the table, Impression_n means the number of the nonclick samples and the Click_num means the number of the click samples.

5.1. The CTR Prediction Model. From the above feature extraction process, we can conclude that the CTR of the RTB advertisement has a great relationship with the users' interest and the basic attributes. It has a little relationship with most of the ID characteristics. Finally, we select the temporal characteristics and the user characteristics like *media_id*, *area_id*, *price_base,* and *price_win* as the input of the prediction model based on the proposed method.

It is necessary to explore the influence of the number of the hidden nodes and the activation function on the speed and the accuracy of the ELM algorithm.

The ELM algorithm provides four kinds of activation functions. From Figure 4, we can know that when the number of the hidden nodes is the same and the activation function is sine function, the AUC value is higher than the other three types of the activation functions about 5%. In addition, the training speed of the sine function is slower than the sigmoid function and the tribas function, but faster than the hardlim function. Considering the training time and the equipment cost, the number of the hidden nodes is set to 500, and the activation function is set to sine function.

Table 5: Overall CTR estimation AUC performance.

	LR	SVM	ELM
1 : 5	0.852	0.846	0.951
1 : 10	0.803	0.818	0.838
1 : 20	0.751	0.761	0.840
1 : 50	0.732	0.712	0.839
1 : 100	0.625	0.639	0.682
1 : 150	0.507	0.510	0.508

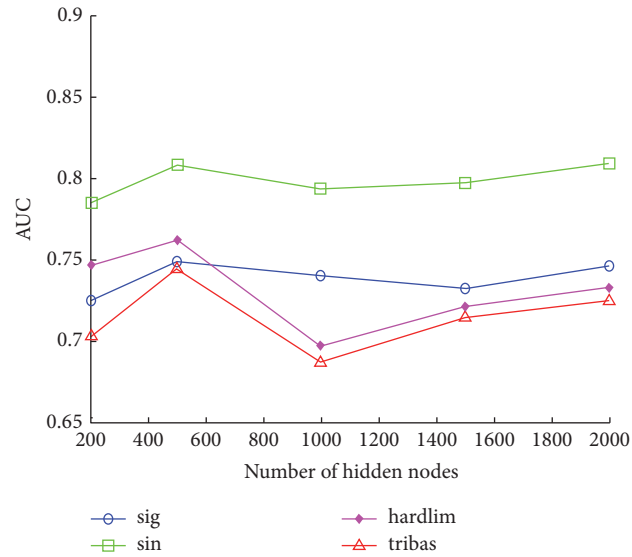


Figure 4: The trend of AUC value with different activation function.

5.2. The Comparison of the Algorithms' Performance. We select logistic regression (LR) model and support vector machine (SVM) model as the comparison methods which are commonly used in other papers, and AUC values of three algorithms are shown in Table 5.

Table 5 shows that the performance of ELM is better than LR and SVM on all the tested datasets, which shows that we have chosen the reasonable characteristics and ELM algorithm is effective as well.

Finally, we selected the traditional ELM algorithm and the Weighted-ELM algorithm as a contrast method when the positive and the negative samples' ratios are set with different proportions; the trend of the AUC results of the three algorithms is shown in Figure 5.

It can be seen from Figure 5 that when the positive and the negative sample ratio is 1 : 5, the three algorithms' AUC values can reach 0.9 or more. When the positive and the negative

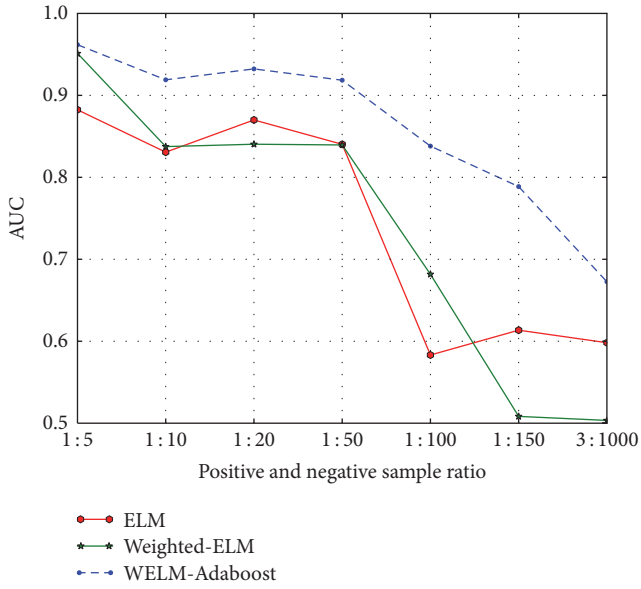


Figure 5: The comparison results of AUC of the three algorithms.

sample ratio is 1 : 50, the AUC value of the WELM-Adaboost algorithm is still above 0.9, but the AUC values of the ELM algorithm and the Weighted-ELM algorithm reduced to 0.84. With the increasing ratio of the sample proportion of the positive and the negative samples, the AUC values of the three algorithms show a decreasing trend, but the AUC value of the WELM-Adaboost algorithm is obviously higher than that of the other two algorithms. The proposed WELM-Adaboost algorithm has a better performance than the other two methods.

The results are shown in Table 6.

For the WELM-Adaboost algorithm, this algorithm has trained 20 Weighted-ELMs as the weak classifier. It can be seen from Table 5 that when the proportion of the positive and the negative samples reaches 1 : 100, the ELM algorithm and the Weighted-ELM algorithm have lower AUC value while the AUC value of the WELM-Adaboost algorithm is still maintained at more than 0.8. This shows that the proposed WELM-Adaboost algorithm has better performance.

6. Conclusions

This paper firstly applied the advertising company's big data to build the user graph system for the purpose of classifying the advertisement data. The output of this user graph system includes the user's age, gender, and the interest preferences, which are used as the input of the prediction model of CTR. Experiments show that this kind of features has a significant effect on the CTR prediction.

The main contribution of the paper is to propose a WELM-Adaboost algorithm based approach for the CTR prediction of the RTB advertisement. We applied the real advertisement dataset to implement the experiments by applying the AUC value as the measurement criteria. We compared both the ELM algorithm and the Weighted-ELM algorithm with the proposed approach. The experimental

Table 6: The Comparison of the AUC value of the three algorithms.

Proportion	ELM	Weighted-ELM	WElM-Adaboost
1 : 5	0.951	0.882	0.962
1 : 10	0.838	0.831	0.919
1 : 20	0.840	0.870	0.932
1 : 50	0.839	0.840	0.919
1 : 100	0.682	0.583	0.838
1 : 150	0.508	0.614	0.789
3 : 1000	0.503	0.569	0.679

results show that the AUC value of the proposed algorithm is significantly improved compared to the ELM and the Weighted-ELM based method.

Although this paper has made a systematic study on the feature extraction and the CTR prediction of the RTB advertisement, there are still some issues to be improved in the future.

The deep neural network may be a good way for the further future study of the CTR prediction.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper is funded by the National Natural Science Foundation of China (nos. 61673056 and 61673055).

References

[1] Z. Meng, *Research on the Personalized Advertising Push Services for Internet Users*, Donghua University, Shanghai, China, 2014.

[2] A. K. Menon, K. Chitrapura, S. Garg, D. Agarwal, and N. Kota, "Response prediction using collaborative filtering with hierarchies and side-information," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '11)*, pp. 141–149, San Diego, Calif, USA, August 2011.

[3] M. Richardson, E. Dominowska, and R. Ragno, "Predicting clicks: estimating the click-through rate for new ads," in *Proceedings of the 16th International World Wide Web Conference (WWW '07)*, pp. 521–530, May 2007.

[4] O. Chapelle, "Modeling delayed feedback in display advertising," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*, pp. 1097–1105, New York, NY, USA, August 2014.

[5] Di. Shao, *Research on High Level Feature Representation and Predicting Methods in Online Advertising*, Harbin Institute Of Technology, Harbin, China, 2014.

[6] T. Fawcett, "ROC graphs: notes and practical considerations for researchers," *Machine Learning*, vol. 31, no. 1, pp. 1–38, 2004.

[7] W. Xiao-Shu, *Click-Through Rate Prediction Based on Deep Neural Netwook Model*, Beijing University Of Posts And Telecommunications, Beijing, China, 2015.

[8] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[9] W. Zong, G.-B. Huang, and Y. Chen, "Weighted extreme learning machine for imbalance learning," *Neurocomputing*, vol. 101, pp. 229–242, 2013.

[10] X. Zhang, Y. Zhuang, W. Wang, and W. Pedrycz, "Transfer boosting with synthetic instances for class imbalanced object recognition," *IEEE Transactions on Cybernetics*, no. 99, pp. 1–14, 2016.

[11] K. Li, X. Kong, Z. Lu, L. Wenyin, and J. Yin, "Boosting weighted ELM for imbalanced learning," *Neurocomputing*, vol. 128, pp. 15–21, 2014.

[12] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 985–990, July 2004.

[13] H.-J. Rong, Y.-S. Ong, A.-H. Tan, and Z. Zhu, "A fast pruned-extreme learning machine for classification problem," *Neurocomputing*, vol. 72, no. 1–3, pp. 359–366, 2008.

[14] Y. Xu, Q. Wang, Z. Wei, and S. Ma, "Traffic sign recognition based on weighted ELM and AdaBoost," *IEEE Electronics Letters*, vol. 52, no. 24, pp. 1988–1990, 2016.

Hindawi
Scientific Programming
Volume 2017, Article ID 6972461, 13 pages
https://doi.org/10.1155/2017/6972461

Research Article

Development of Multiple Big Data Analytics Platforms with Rapid Response

Bao Rong Chang, Yun-Da Lee, and Po-Hao Liao

Department of Computer Science and Information Engineering, National University of Kaohsiung, 700 Kaohsiung University Rd., Nanzih District, Kaohsiung 811, Taiwan

Correspondence should be addressed to Bao Rong Chang; brchang@nuk.edu.tw

Received 6 April 2017; Accepted 28 May 2017; Published 21 June 2017

Academic Editor: Wenbing Zhao

The crucial problem of the integration of multiple platforms is how to adapt for their own computing features so as to execute the assignments most efficiently and gain the best outcome. This paper introduced the new approaches to big data platform, RHhadoop and SparkR, and integrated them to form a high-performance big data analytics with multiple platforms as part of business intelligence (BI) to carry out rapid data retrieval and analytics with R programming. This paper aims to develop the optimization for job scheduling using MSHEFT algorithm and implement the optimized platform selection based on computing features for improving the system throughput significantly. In addition, users would simply give R commands rather than run Java or Scala program to perform the data retrieval and analytics in the proposed platforms. As a result, according to performance index calculated for various methods, although the optimized platform selection can reduce the execution time for the data retrieval and analytics significantly, furthermore scheduling optimization definitely increases the system efficiency a lot.

1. Introduction

Big data [1] has been sharply in progress unprecedentedly in recent years and is changing the operation for business as well as the decision-making for the enterprise. The huge amounts of data contain valuable information, such as the growth trend of system application and the correlation among systems. The undisclosed information may contain unknown knowledge and application that are discoverable further. However, big data with the features of high volume, high velocity, and high variety as well as in face of expanding incredible amounts of data, several issues emerging in big data such as storage, backup [2], management, processing, search [3], analytics, practical application, and other abilities to deal with the data also face new challenges. Unfortunately, those cannot be solved with traditional methods and thus it is worthwhile for us to continue exploring how to extract the valuable information from the huge amounts of data. According to the latest survey reported from American CIO magazine, 70% of IT operation has been done by batch processing in the business, which makes it "unable to control processing resources for operation as well as loading" [4]. This becomes one of the biggest challenges for big data application.

Hadoop distributes massive data collections across multiple nodes, enabling big data processing and analytics far more effectively than was possible previously. Spark, on the other hand, does not do distributed storage [5]. It is nothing but a data processing tool, operating on those distributed data collections. Furthermore, Hadoop includes not only a storage component called Hadoop Distributed File System (HDFS), but also a processing component called MapReduce. Spark does not come with its own file management system. Accordingly, it needs to be integrated with Hadoop to share HDFS. Hadoop processing mostly static and batch-mode style can be just fine and originally was designed to handle crawling and searching billions of web pages and collecting their information into a database [6]. If you need to do analytics on streaming data, or to run required multiple operations, Spark is suitable for those. As a matter of fact,

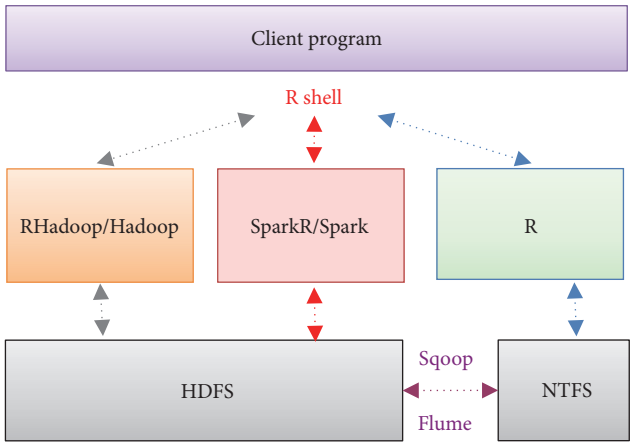


Figure 1: Data retrieval and data analytics Stack.

Spark was designed for Hadoop; therefore, data scientists all agree they are better together for a variety of big data applications in the real world.

Through establishing a set of multiple big data analytics platforms with high efficiency, high availability, and high scalability [7], this paper aims to integrate different big data platforms to achieve the compatibility with any existing business intelligence (BI) [8] together with related analytics tools so that the enterprise needs not change large amounts of software for such platforms. Therefore, the goal of this paper is to design the optimization for job scheduling using MSHEFT algorithm as well as to implement optimized platform selection, and established platforms support R command to execute data retrieval and data analytics in big data environment. In such a way the upper-level tools relying on relational database which has stored the original data can run on the introduced platforms through minor modification or even no modification to gain the advantages of high efficiency, high availability, and high scalability. I/O delay time can be shared through reliable distributed file system to speed-up the reading of a large amount of data. Data retrieval and data analytics stack has layered as shown in Figure 1. As a result, according to performance index calculated for various methods, we are able to check out whether or not the proposed approach can reduce the execution time for the data retrieval and analytics significantly.

2. Related Work in Big Data Processing

This paper has introduced data retrieval and data analytics using R programming in conjunction with RHadoop [9]/Hadoop [10] and SparkR [11]/Spark [12] platforms to build a multiple-platform big data analytics system. Furthermore, the use of distributed file system for fast data analytics and data storage reduces the execution time of processing a huge amount of data. First let us aim to understand the fundamental knowledge of Hadoop and Spark platforms and then build their extended systems RHadoop and SparkR for the purpose of fitting all kinds of relative problems on big data analytics. This section will introduce their related profiles and key technologies for both platforms accordingly.

2.1. Distributed Computing Framework with Hadoop. Hadoop is a well-known open source distributed computing framework as shown in Figure 2 that provides reliable, scalable, distributed computing, data storage, and cluster computing analytics of big data, including a MapReduce [13] for distributed computing, HDFS [14] distributed file system, and a distributed NoSQL database HBase [15] which can be used to store nonrelational data set. There are some tools that are based on Hadoop applications. First Apache Pig can perform complex MapReduce conversions on a huge amount of data using a simple scripting language called Pig Latin. Next Apache Hive [16] is a data warehousing package that lets you query and manage large datasets in distributed storage using a SQL-style language called HiveQL. Third Apache Sqoop is a tool for transferring large amounts of data between Hadoop and structured data storage as efficiently as possible. Further Apache Flume is a distributed and highly scalable log collection system that can be used for log data collection, log data processing, and log data transmission. Then Apache Zookeeper is a distributed application designed for the coordination of services, it is mainly used to solve the decentralized applications often encountered in some data management issues. Final Apache Avro is a data serialization system designed to support intensive data, the application of huge amounts of data exchange.

Examples of applications using Hadoop are given as follows. Caesars entertainment, a casino gaming company, has built a Hadoop environment [17] that differentiates customer groups and creates exclusive marketing campaign for each group. Healthcare technology company Cerner uses Hadoop to build a set of enterprise data centers [18] to help Cerner and their clients monitor the health of more than one million patients a day. The dating site eHarmony uses Hadoop to upgrade their cloud systems [19], enabling it to send millions of messages for matching friend dating every day.

2.2. Parallel Processing Framework with Spark. Spark is an open source parallel processing framework released by the Apache Software Foundation that supports in-memory processing and dramatically increases the execution speed of big data analytics, as shown in Figure 3. Spark is also designed for fast computing, high availability, and fault tolerance. Using its internal memory capabilities, Spark can be a great choice for machine learning and graph computation, as well as a great choice for big data analytics. Its main functions and positioning are the same as Hadoop MapReduce. Through In-memory cluster computing [20], it hopes to eliminate I/O latency caused by a lot of relay files swapped between memory and disk during MapReduce. Theoretically, the processing speed could be hundreds of times higher than the Hadoop. Spark is written in Scala, but also supports Scala, Java, and Python programming; the underlying storage system can also be directly compatible with HDFS.

Examples of Spark's applications are given as follows. Microsoft launched Spark for Azure HDInsight [21], allowing

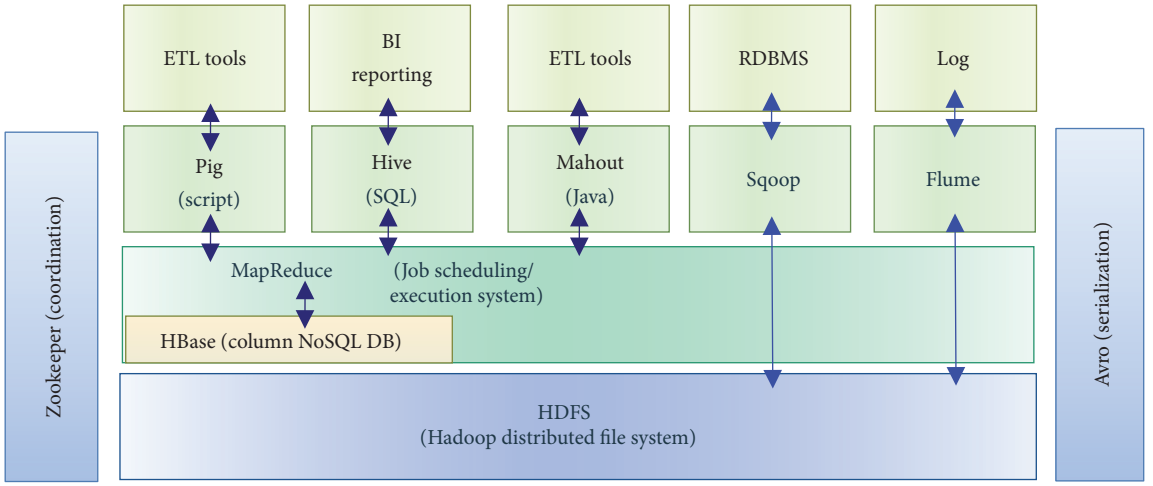


Figure 2: Hadoop framework.

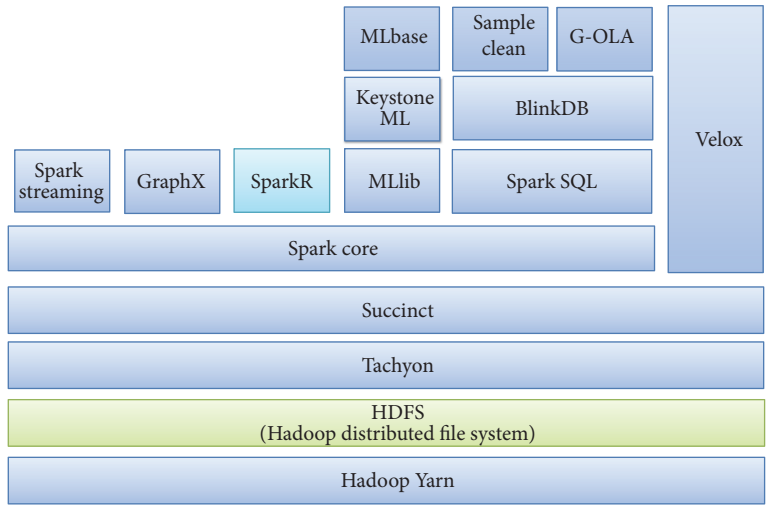


Figure 3: Spark framework.

users to use Spark for Azure HDInsight to solve big data challenges in near real-time, such as fraud detection, clickstream analytics, financial alerts, and more. Yahoo used Spark for the development of Audience Expansion in the application of advertising [22] to find the target user. Cloudera develops Spark Streaming's flexibility [23] to enable Cloudera's customers to build complete IoT applications on a unified platform.

2.3. Integrated Development Environment for R. Over the past decade, programming language R has been highly enhanced and greatly upgraded significantly to break the original limit in the past. In academy and industry, R becomes one of the most important tools for the research such as computational statistics, visualization, and data science. Millions of statisticians and data scientists use R to solve problems from counting biology to quantitative marketing. R has become one of the most popular programming language for the analytics of scientific data and finance. R is not only free, compact, and part of the open source that can run on many platforms, but also integrates data analytics and plotting functions all in one. It may add many additional packages to enhance system's functions, similarly comparable to the functions of commercial software, and can be viewed as one of major tools of contemporary data analytics. R is mainly used to analyze data, and thus the master node in a cluster installs R where big data access through HDFS has been available, or a stay alone computer for centralized processing installs R where small data access through NTFS has achieved. It is noted that data stored in NTFS can be transferred to HDFS via Sqoop [24]/Flume [25] or Hive.

2.4. RHadoop Based on Hadoop. Hadoop is capable of distributed computing and can store large amounts of data, but there is still a lot of information that needs to be analyzed professionally. However, R itself is not able to read the data size more than the size of memory in computer, and hence there is data size limit for processing big data. Therefore, it

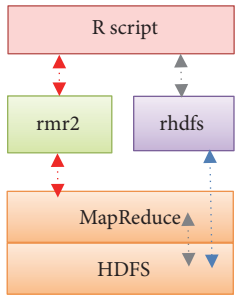


Figure 4: RHadoop framework.

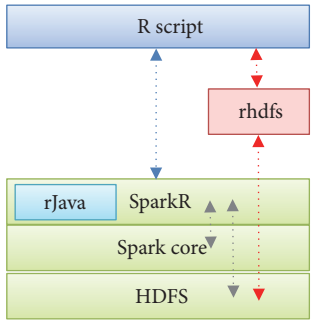


Figure 5: SparkR framework.

turns out the integration of Hadoop and R called RHadoop as a sort of data analytics service. In such a way, R will not only handle professional analytics, but it will also allow to easily utilize Hadoop features, such as the ability to access HDFS via rhdfs package and through the rmr2 package [26] to call MapReduce for accomplishing the distributed computing. The framework of RHadoop is shown in Figure 4.

2.5. SparkR Based on Spark. SparkR is an R suite developed by AMPLab that provides Spark with a Resilient Distributed Dataset (RDD) [27] API that allows R to carry out distributed computing using Spark. SparkR was merged into Spark in April 2015 and was released with Spark 1.4 in June 2015, so deploying SparkR requires installing Spark 1.4 or later and installing R related packages, including rJava [28] and rhdfs [29]. rJava lets R call objects, instances, and methods written in Java to make it less difficult for R to call Java-owned resources, such as Spark and Hadoop, and rhdfs, like RHadoop, to access HDFS. The framework of SparkR is shown in Figure 5. Although RHadoop mentioned above can activate distributed computing with R programming, its efficiency is not as good as SparkR. SparkR, adopting in-memory cluster computing, needs more memory resources than RHadoop. In order to avoid shutting down the task due to hardware resources limitation, both RHadoop and SparkR can be installed together for being interchangeably used at same site. In addition, in order to determine the most suitable analytical tools, we also need a matching algorithm to carry out the distributed computing successfully.

Table 1: Recipe of compatibility packages.

Software	Version
Hadoop (including RHadoop)	2.6.0
Spark (including SparkR)	1.4.0
R	3.2.2
Oracle Java (JDK)	8u66
Scala	2.10.4
rJava	0.9.7
rhdfs	1.0.8
rmr2	3.3.1

3. System Implementation Method

This paper aims to develop the optimization for job scheduling using MSHEFT algorithm so that system obtains the best throughput. After scheduling all of input queries in a job queue, system is then able to dispatch the job at top of the queue to one of big data analytics platforms through automatic platform selection. Regarding clustering and distributed parallel computing, a cloud computing foundation has been established to implement virtualization architecture because virtual machine has the feature of flexible control in hardware resource and thus it is quite suitable to act as a container provided an environment for the exploration of big data analytics.

3.1. Virtual Machine Deployment. Figure 6 shows a cloud computing [30] with high performance, high availability, and high scalability where server farm at the top layer and storage farm at the bottom layer are built for this study. In order to realize virtualization, an open source virtual machine management (VMM) or hypervisor Proxmox Virtual Environment (PVE) [31] based on KVM is used to implement virtual machine clustering; the status of virtual machine clustering can be effectively monitored through PVE, and the resource configuration of each virtual machine can be dynamically adjusted [32]. Since the platform performance is very closely related to I/O latency, the efficiency of both hard disk and network access should be increased in hardware configuration.

3.2. Recipe of Compatibility Packages. The most difficult aspect of integration of a lot of open source packages in a system is compatibility suite and that is one of the crucial problems of system integration as well. In this paper we proposed a recipe to resolve the challenge of suite compatibility. Several packages will be integrated to establish multiple big data analytics platforms in this paper and all of them are open source software, which are developed and maintained by different open source communities. A lot of software has complex dependency and compatibility problems. The recipe of packages proposed in this paper includes Hadoop, Spark, R, Scala, rJava, rhdfs, and rmr2, which are fully compatible for stable operation in the proposed approach as listed in Table 1.

3.3. Optimized Platform Selection. The program of automatic platform selection assigns a task to an appropriate big data

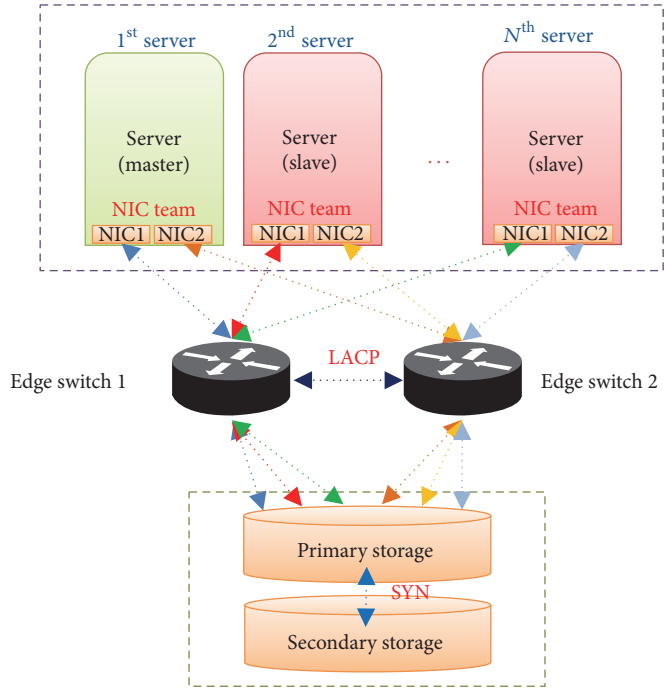


Figure 6: Cloud computing with high performance, high availability, and high scalability.

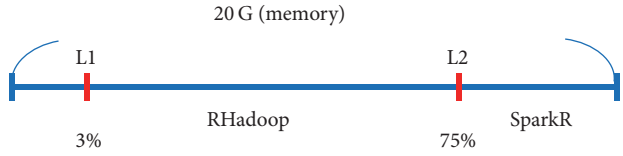


Figure 7: Automatic selection of suitable platform.

analytics platform according to the size of remaining amount of memory in a working virtual machine. The function and property for both RHadoop and SparkR are identical in a sense because they can access the same HDFS and support R syntax. Although these two platforms are the same function, they are different in the demand environment and executive manner. The memory size of 20 G for each server in the experiments is given, and it sets the remaining amount of memory size 0.6 G in a virtual machine in cluster denoted Level 1 (roughly 3% of total amount of memory) and 15 G Level 2 (approximately 75% of total amount of memory) as the cut-off points. In Figure 7, the program automatically chooses nothing to carry on the task as the remaining amount of memory is less than 3%; RHadoop would be applied as the remaining amount of memory lies between L1 and L2, and hence SparkR could be employed as the remaining amount of memory is higher than L2.

3.4. Optimization for Job Scheduling. Heterogeneous Earliest Finish Time (HEFT) [33] is an exploratory scheduling algorithm, which is used for scheduling the communication time of previous set of dependent task of heterogeneous network. HEFT is based on one of list scheduling algorithms, where their characteristics are to establish a priority list in the first step. According to the sorted priority list, HEFT assigns each task to a suitable CPU to make the task completed as soon as possible. The pseudocode of HEFT algorithm is shown in Algorithm 1. HEFT tries to search for local optimization and eventually makes the whole local optimums. In the test of automatic platform selection, the total of 20 GB memory is configured, and it is found that all of analytics platforms can be used when the remaining amount of memory is greater than or equal to L1; in addition, it is better to use RHadoop in case of being less than L2, and SparkR shall be used in case of being greater than L2. Job dispatched to RHadoop platform has run a kind of in-disk computing mode such that it may encounter data swap between disk and memory occasionally. Instead, in-memory computing mode has employed in SparkR platform and thus SparkR needs much more memory allocated for computing. HEFT algorithm is modified to Memory-Sensitive Heterogeneous Earliest Finish Time (MSHEFT) where the priority is considered first; then the size of data file is considered as the second condition, and finally an extra factor is considered, which is "remaining amount

(1) Compute rank_u for all nodes by traversing graph upward, starting from the exit node.
(2) Sort the nodes in a list by nonincreasing order of rank_u values.
(3) while there are unscheduled nodes in the list do
(4) begin
(5) Select the first task n_i in the list and remove it.
(6) Assign the task n_i to the processor p_j that minimizes the (EFT) value of n_i.
(7) end

Algorithm 1: The HEFT algorithm.

(1) Compute rank_u for all nodes by traversing graph upward, starting from the exit node.
(2) Sort the nodes in a list by nonincreasing order of rank_u values.
(3) while there are unscheduled nodes in the list do
(4) Compare priority.
(5) begin
(6) Compare job size
(7) Select the first task n_i in the list and remove it.
(8) begin
(9) if the remaining memory size > 0.6 GB
(10) begin
(11) what is the value of the remaining memory size?
(12) Assign the task n_i to the processor p_j that minimizes the (EFT) value of n_i.
(13) end if
(14) waiting the remaining memory size and go line 9.
(15) end
(16) end

Algorithm 2: The MSHEFT algorithm.

of memory." In Algorithm 2, the pseudocode of MSHEFT algorithm has been presented. Job processing flow chart is shown in Figure 8.

3.5. Execution Procedure. The execution procedure has been shown in Figure 9. With the user interface, the process is designated to monitor the status of each node in the server farm. MSHEFT algorithm for scheduling optimization together with platform selection has decided to choose an appropriate platform for execution according to the current status monitored through user interface. The proposed approach including MSHEFT algorithm plus platform selection can be denoted MSHEFT-PS in this paper. When the analytics task has finished, the results will be stored back to HDFS and the whole process will be terminated. In addition, job scheduling using first-come-first-serve FCFS will be adopted for each single analytics platform Rhadoop or SparkR, denoted FCFS-SP, in the experiment to check how it performs as a single platform applied. Furthermore, the platform selection mechanism integrated FCFS, denoted FCFS-PS, has also been employed to test the system performance under the condition of remaining amount of memory in a virtual machine in which a certain node has been resident.

3.6. Performance Evaluation. In order to compare the computation efficiency among the several algorithms, the performance index [2] has been evaluated based on the necessitated equations, which are derived first from measuring access time of data of a single item for a certain dataset on (1), next calculating average access time based on a variety of data size among the datasets on (2), then inducing a normalized performance index among the datasets on (3), and finally resulting in a performance index according to a series of tests on (4). In these equations we denote the subscript i the index of data size, j the index of dataset, and k the index of test condition and the subscript s indicates a single item in a specific dataset. Eq. (1) calculates the average access time (AAT) for each data size. In (1), AAT_{ijk} represents average access time with the same data size, and N_{ik} stands for the current data size. Eq. (2) calculates the average access times overall

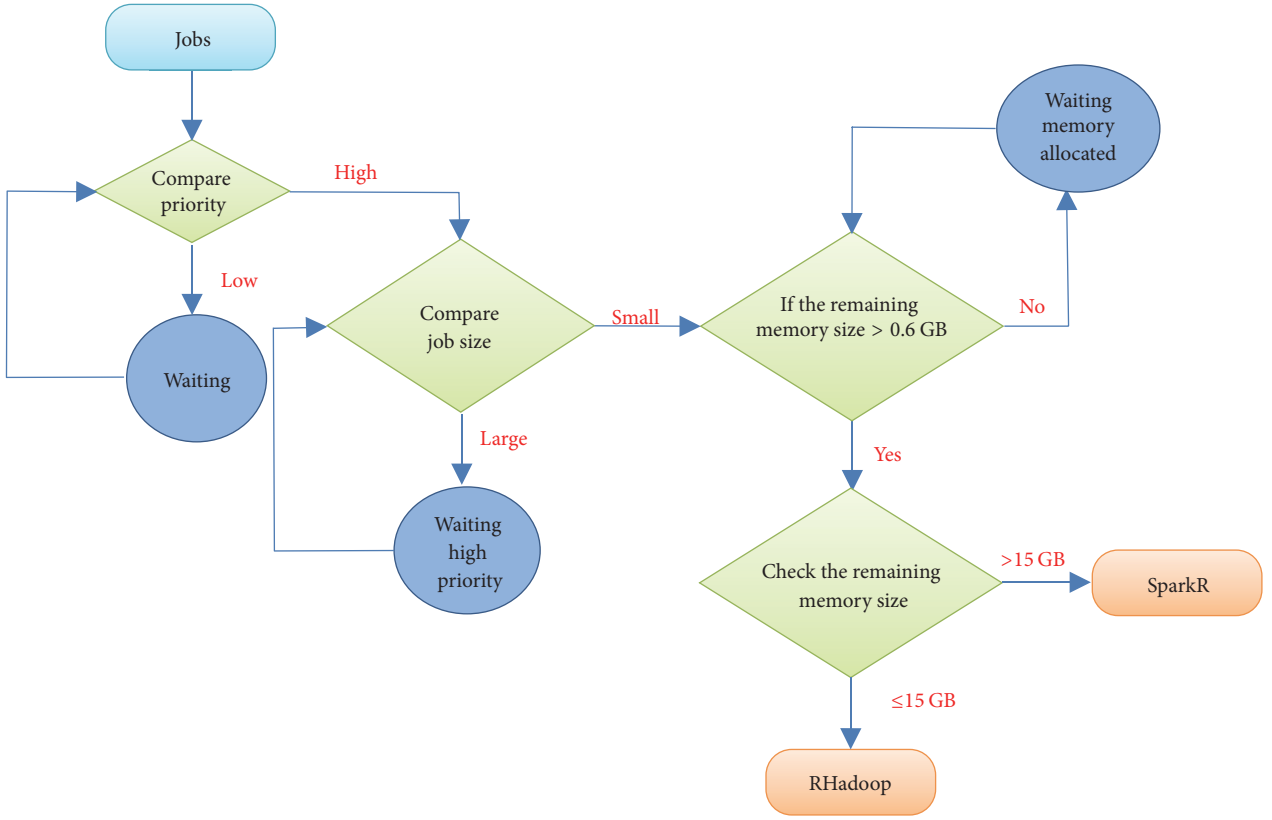


Figure 8: Job processing flow chart with MSHEFT algorithm and platform selection.

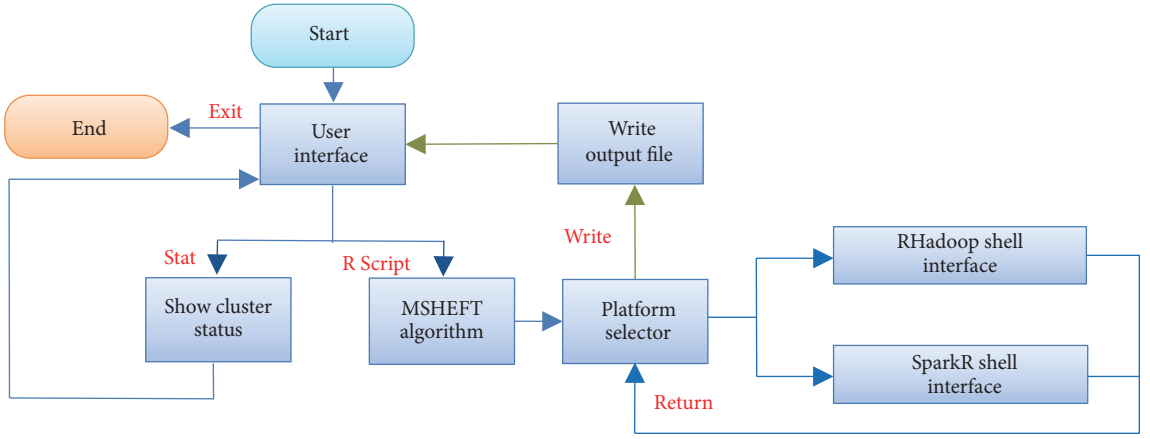


Figure 9: Execution procedure flow chart.

$\overline{\mathrm{AAT}_{s_{jk}}}$ for each test (i.e., write, read, and compute) on a specific platform, in which $\mathrm{AAT}_{s_{ijk}}$ represents the average access time of each dataset; please refer back to (1), and ω_i stands for weight for a weighted average. The following formula will evaluate the performance index (PI) [10]. Eq. (3) calculates the normalized performance index for a specific platform. Eq. (4) calculates the performance index overall for a specific platform, SF_1 is a constant value that is used here to quantify the value of performance index in the range 0–100, and W_k stands for weight for a weighted average.

$$\mathrm{AAT}_{s_{ijk}} = \frac{\mathrm{AAT}_{ijk}}{N_{ik}}, \quad \text{where } s = 1, 2, \ldots, d;\ i = 1, 2, \ldots, l;\ j = 1, 2, \ldots, m;\ k = 1, 2, \ldots, n, \tag{1}$$

$$\overline{\mathrm{AAT}_{s_{jk}}} = \sum_{i=1}^{l} \omega_i \cdot \mathrm{AAT}_{s_{ijk}}, \quad \text{where } s = 1, 2, \ldots, d;\ j = 1, 2, \ldots, m;\ k = 1, 2, \ldots, n;\ \sum_{i=1}^{l} \omega_i = 1, \tag{2}$$

$$\overline{\mathrm{PI}_{jk}} = \frac{1/\overline{\mathrm{AAT}_{s_{jk}}}}{\max_{h=1,2,\ldots m}\left(1/\overline{\mathrm{AAT}_{s_{hk}}}\right)}, \quad \text{where } j = 1, 2, \ldots, m;\ k = 1, 2, \ldots, n, \tag{3}$$

$$\mathrm{PI}_j = \left(\sum_{k=1}^{n} W_k \cdot \overline{\mathrm{PI}_{jk}}\right) \cdot \mathrm{SF}_1, \quad \text{where } j = 1, 2, \ldots, m;\ k = 1, 2, \ldots, n;\ \mathrm{SF}_1 = 10^2,\ \sum_{k=1}^{n} W_k = 1. \tag{4}$$

4. Experimental Results and Discussion

This section categories data into simulation data and actual data for test with two cases; the first case (Case 1) uses the test data generated randomly with Java programming; the second case (Case 2) adopts the actual data collected from the Internet. Proxmox Virtual Environment can be used to dynamically adjust the resource allocation to set up the experimental environments according to different memory remaining amounts, as listed in Table 2, so as to implement effect tests on various platforms.

4.1. Generated Data Set and Experimental Environment for Case 1. Case 1 tests each platform with first-come-first-serve algorithm to perform different sizes of test data, R commands having different complexity, and different priorities to all of queries so as to compare the execution time in various environments as is shown in Table 2. R commands for test are as shown in Table 3. In this experiment, there are three methods applied to test. The first approach uses first-come-first-serve algorithm (FCFS) for each single platform RHaoop or SparkR, denoted FCFS-SP. The second one is an optimized platform selection (PS) utilized to choose an appropriate platform for execution according to the remaining amount of memory in a virtual machine but it is still based on FCFS, thus denoted FCFS-PS. The third method introduced the optimization for job scheduling using MSHEFT algorithm employed to reschedule all of input queries in an ascending order in a job queue according to the smallest size of data file first. Once a job has been dequeued and launched, it based on PS will also choose an appropriate platform for execution, thereby denoted MSHEFT-PS. In short, three approaches including FCFS-SP, FCFS-PS, and MSHEFT-PS will be implemented in this paper. The test methods are shown in Table 4. With four fields, test data have been randomly generated with Java programming where the first column is the name of the only key string, the second column is random integer from 0 to 99, the third column is a random integer from 100 to 199, and the fourth column is the generated integer sequence number. Designated data size for test is shown in Table 5.

4.2. Experimental Results in Case 1. As a result, two platforms, RHadoop and SparkR, have performed for several test data sets with different priorities, data sizes, and R commands. As listed in Table 6, the proposed approach MSHEFT-PS has been implemented in the different order of jobs in a queue when comparing with the other methods. Performance comparisons of test are shown in Figures 10, 11, 12, 13, 14, and 15. The average execution time of proposed approach MSHEFT-PS is faster than the other methods, FCFS-SP and FCFS-PS. The normalized performance index and performance index are listed in Tables 7 and 8. This shows that the proposed approach outperforms the others in Case 1.

4.3. Data Collection and Experimental Environment for Case 2. Case 2 has collected the actual data sets and the designated data size for test as shown in Table 9. The concerned approaches as listed in Table 3 have applied for measuring the average execution time according to different data themes. Similarly, Table 1 has listed two test environments and R command I test is listed in Table 2 as well.

4.4. Experimental Results in Case 2. Executable job list in Case 2 is shown in Table 10. Performance comparisons of test are shown in Figures 16 and 17. The experimental results show that the average execution time of the proposed approach MSHEFT-PS is much lower than the other methods, FCFS-SP and FCFS-PS, over three different conditions. The normalized performance index and performance index are listed in Table 11. Notice that the performance of our proposed approach is superior to the others in Case 2.

4.5. Discussion. There is no specific mechanism so far to estimate job execution time for Rhadoop or SparkR. According to the report in Apache Spark website at https://spark.apache.org/, it noted that run programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk. Technically speaking, SparkR job execution will similarly be up to 100x faster than RHadoop job execution in-memory, or up to 10x on disk as mentioned above. In this paper, the experiments show that run program for a specific job using SparkR is up to 9.7x faster than RHadoop. However, the average in SparkR job execution is nearly 3.9x faster than RHadoop job execution.

5. Conclusion

This paper found that even though the analytics platforms have the same configuration and functions, their performance still has resulted in different efficiency in different experimental conditions when applying scheduling optimization for multiple big data analytics platforms. The performance efficiency can be greatly improved by making the optimization for job scheduling, automatically detecting clustering state, and then choosing an appropriate platform for job processing. According to the experiments in Case 1 with simulation data and Case 2 with actual data, it is found that the remaining amount of memory is less and the scale of

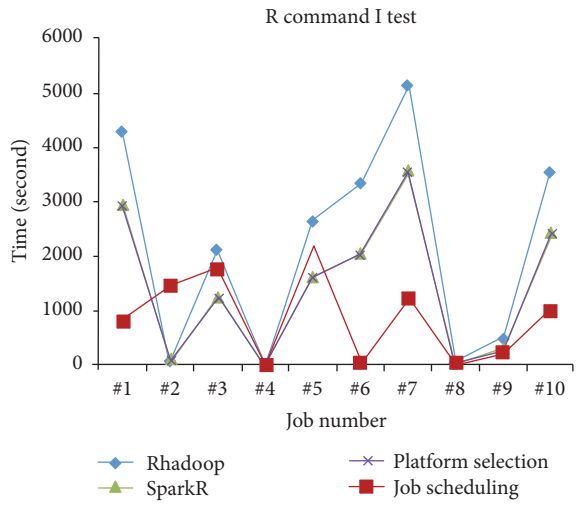


Figure 10: Execution time of R command I in test environment I of Case 1.

Table 2: Test environment.

Environment	Description
Test environment I	Adjust 10 GB memory space and give it to a virtual machine executing big data processing
Test environment II	Configure 20 GB memory space of a virtual machine executing big data processing

Table 3: R command test.

Command	Description
R command I	Only search special field
R command II	Only search special field, and add comparison conditions
R command III	Search special field, add comparison conditions, and execute the commands with while or for

Table 4: Test method.

Method	Description
FCFS-SP	Use command of enforced R to execute such platform, and then input R command
FCFS-PS	Directly input R command
MSHEFT-PS	Use command of set to set working quantity, and then input R command

data set is larger, which will much more highlight the importance of scheduling optimization and platform selection. In addition to the job scheduling using MSHEFT algorithm and optimized platform selection proposed in this paper, this system is capable of integrating new analytics platform to it by adding new big data analytics tools with related R shells to system, without any further changes in others.

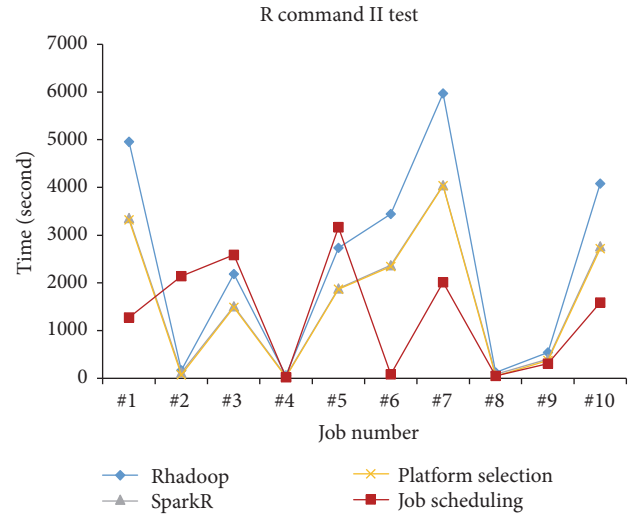


Figure 11: Execution time of R command II in test environment I of Case 1.

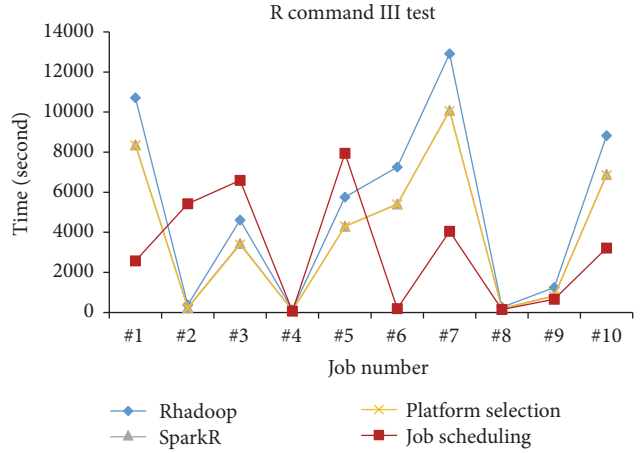


Figure 12: Execution time of R command III in test environment I of Case 1.

Table 5: Designated data size and its priority in Case 1.

Sequence	Priority	Data size	Code name
1	1	850 G	A
2	3	30 G	B
3	1	400 G	C
4	2	10 G	D
5	5	500 G	E
6	3	630 G	F
7	2	1 T	G
8	4	20 G	H
9	5	100 G	I
10	1	700 G	J

Table 6: Executable job list in Case 1.

Method	Job									
	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
FCFS-SP	A	B	C	D	E	F	G	H	I	J
FCFS-PS	A	B	C	D	E	F	G	H	I	J
MSHEFT-PS	C	J	A	D	G	B	F	H	I	E

Table 7: Normalized performance index in Case 1.

Operation	FCFS-SP-RHadoop	FCFS-SP-SparkR	FCFS-PS	MSHEFT-PS
R command I	0.319	0.787	0.799	1.000
R command II	0.441	0.884	0.895	1.000
R command III	0.481	0.880	0.885	1.000

Table 8: Average normalized performance index and performance index in Case 1.

Method	Average normalized performance index	Performance index
FCFS-SP-RHadoop	0.413	41.34
FCFS-SP-SparkR	0.850	85.03
FCFS-PS	0.859	85.94
MSHEFT-PS	1.000	100.00

Table 9: Designated data size and its priority in Case 2.

Sequence	Priority	Data size	Data theme	Code name
1	4	10 G	World-famous masterpiece	WC
2	1	250 G	Load of production machine: Overlaoding	OD
3	2	250 G	Load of production machine: Underloading	UD
4	3	1 T	Qualified rate of semiconductor products	YR
5	1	750 G	Correlation among temperature and people's power utilization	TE
6	4	750 G	Correlation among rainfall and people's power utilization	RE
7	1	100 G	Flight information in the airport	AP
8	5	500 G	Traffic violation/accidents	TA

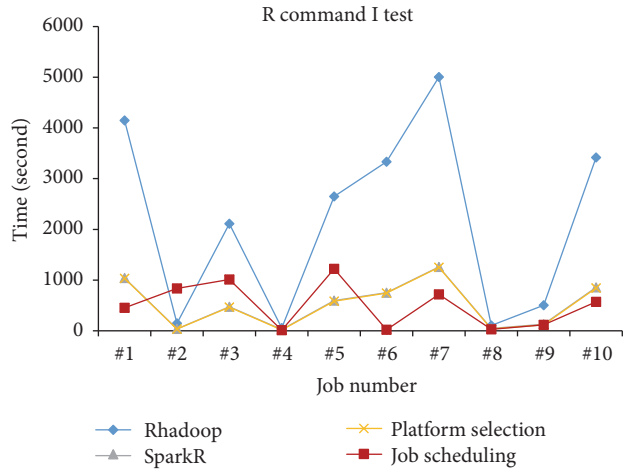


Figure 13: Execution time of R command I in test environment II of Case 1.

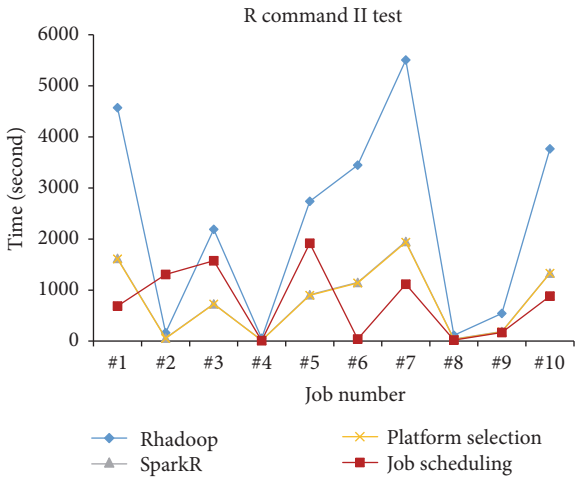


Figure 14: Execution time of R command II in test environment II of Case 1.

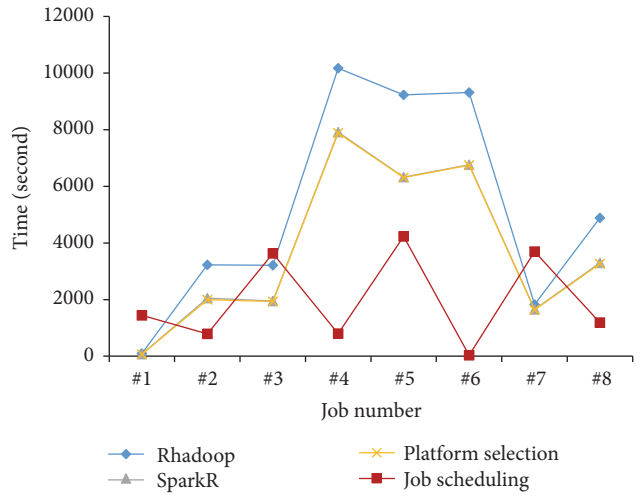


Figure 16: Execution time of experimental environment I in Case 2.

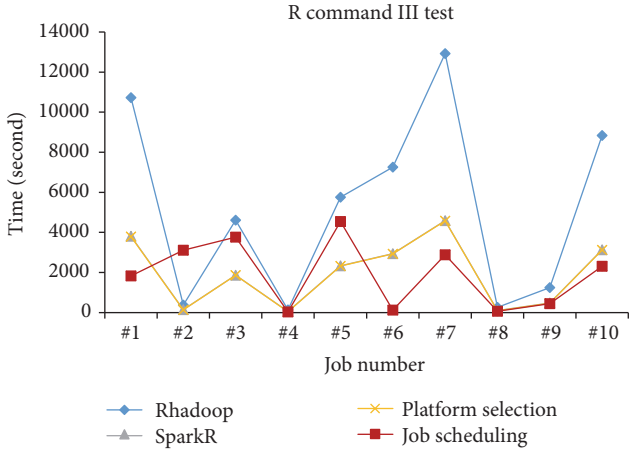


Figure 15: Execution time of R command III in test environment II of Case 1.

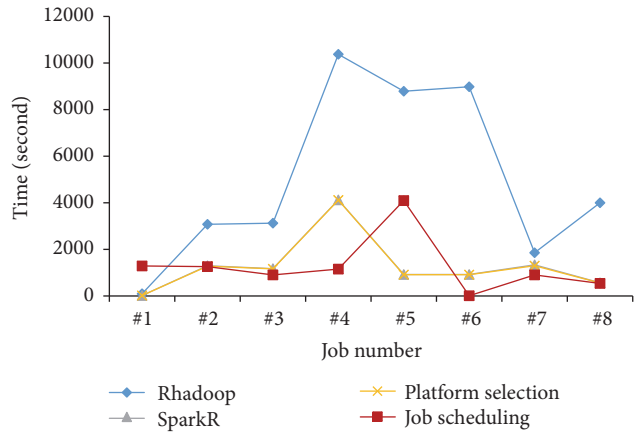


Figure 17: Execution time of experimental environment II in Case 2.

Table 10: Executable job list in Case 2.

Method	Job							
	#1	#2	#3	#4	#5	#6	#7	#8
FCFS-SP	WC	OD	UD	YR	TE	RE	AP	TA
FCFS-PS	WC	OD	UD	YR	TE	RE	AP	TA
MSHEFT-PS	AP	OD	TE	UD	YR	WC	RE	TA

Table 11: Average normalized performance index and performance index in Case 2.

Method	Average normalized performance index	Performance index
FCFS-SP-RHadoop	0.314	31.42
FCFS-SP-SparkR	0.753	75.32
FCFS-PS	0.760	76.02
MSHEFT-PS	1.000	100.00

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is fully supported by the Ministry of Science and Technology, Taiwan, under Grant nos. MOST 105-2221-E-390-013-MY3 and MOST 104-2622-E-390-006-CC3.

References

[1] H. Chen, R. H. L. Chiang, and V. C. Storey, "Business intelligence and analytics: from big data to big impact," *MIS Quarterly: Management Information Systems*, vol. 36, no. 4, pp. 1165–1188, 2012.

[2] B. R. Chang, H.-F. Tsai, and C.-L. Guo, "High performance remote cloud data center backup using NoSQL database," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 7, no. 5, pp. 993–1005, 2016.

[3] B.-R. Chang, H.-F. Tsai, and H.-T. Hsu, "Secondary index to Big Data NoSQL Database—Incorporating solr to HBase approach," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 7, no. 1, pp. 80–89, 2016.

[4] C. D. Wickens, "Processing resources in attention dual task performance and workload assessment," 1981, Office of Naval Research Engineering Psychology Program, No. N-000-14-79-C-0658.

[5] P. Mika and G. Tummarello, "Web semantics in the clouds," *IEEE Intelligent Systems*, vol. 23, no. 5, pp. 82–87, 2008.

[6] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: cluster computing with working sets," in *Proceedings of the 2nd USENIX Workshop on Hot Topics in Cloud Computing*, pp. 95–101, Portland, Ore, USA, 2010.

[7] B.-R. Chang, H.-F. Tsai, Y.-C. Tsai, C.-F. Kuo, and C.-C. Chen, "Integration and optimization of multiple big data processing platforms," *Engineering Computations (Swansea, Wales)*, vol. 33, no. 6, pp. 1680–1704, 2016.

[8] S. Chaudhuri, U. Dayal, and V. Narasayya, "An overview of business intelligence technology," *Communications of the ACM*, vol. 54, no. 8, pp. 88–98, 2011.

[9] D. Harish, M. S. Anusha, and K. V. Daya Sagar, "Big data analytics using RHadoop," *International Journal of Innovative Research in Advanced Engineering*, vol. 2, no. 4, pp. 180–185, 2015.

[10] M. Adnan, M. Afzal, M. Aslam, R. Jan, and A. M. Martinez-Enriquez, "Minimizing big data problems using cloud computing based on Hadoop architecture," in *Proceedings of the 2014 11th Annual High Capacity Optical Networks and Emerging/Enabling Technologies (Photonics for Energy), HONET-PfE 2014*, pp. 99–103, Charlotte, NC, USA, 2014.

[11] X. Yang, S. Liu, K. Feng, S. Zhou, and X.-H. Sun, "Visualization and adaptive subsetting of earth science data in HDFS: a novel data analytics strategy with hadoop and spark," in *Proceedings of the 2016 IEEE International Conferences on Big Data and Cloud Computing, Social Computing and Networking, Sustainable Computing and Communications*, pp. 89–96, Atlanta, Ga, USA, 2016.

[12] Apache Spark, 2017, https://spark.apache.org/.

[13] M. Maurya and S. Mahajan, "Performance analysis of MapReduce programs on Hadoop cluster," in *Proceedings of the 2012 World Congress on Information and Communication Technologies, WICT 2012*, pp. 505–510, Trivandrum, India, 2012.

[14] A. Kala Karun and K. Chitharanjan, "A review on hadoop—HDFS infrastructure extensions," in *Proceedings of the 2013 IEEE Conference on Information and Communication Technologies, ICT 2013*, pp. 132–137, Tamil Nadu, India, 2013.

[15] L. George, *HBase: The Definitive Guide: Random Access to Your Planet-Size Data*, O'Reilly Media, Inc, Sebastopol, Calif, USA.

[16] A. Thusoo, J. S. Sarma, N. Jain et al., "Hive: a warehousing solution over a map-reduce framework," *Proceedings of the VLDB Endowment*, vol. 2, no. 2, pp. 1626–1629, 2009.

[17] Caesars Entertainment, 2017, https://www.cloudera.com/about-cloudera/press-center/press-releases/2015-05-05-cloudera-intel-accelerate-enterprise-hadoop-adoption-industry-partnership.html.

[18] Cerner, 2017, https://www.cloudera.com/customers/cerner.html.

[19] eharmony, 2017, http://www.eharmony.com/engineering/mapping-love-with-hadoop/#.WKCRgTt9600.

[20] J. Heinrich and B. Broeksema, "Big data visual analytics with parallel coordinates," in *Proceedings of the Big Data Visual Analytics, BDVA 2015*, Tasmania, Australia, 2015.

[21] Azure HDInsight, 2017, https://azure.microsoft.com/zh-tw/services/hdinsight/.

[22] G. Li, J. Kim, and A. Feng, "Yahoo audience expansion: migration from hadoop streaming to spark," in *Proceedings of the Spark Summit 2013*, San Francisco, Calif, USA, 2013, Yahoo, 2017, https://spark-summit.org/2013/li-yahoo-audience-expansion-migration-from-hadoop-streaming-to-spark/.

[23] Cloudera Spark Streaming, 2017, https://blog.cloudera.com/blog/2016/05/new-in-cloudera-labs-envelope-for-apache-spark-streaming/.

[24] M. S. Aravinth, M. S. Shanmugapriyaa, M. S. Sowmya, and M. E. Arun, "An efficient hadoop frameworks sqoop and ambari for big data processing," *International Journal for Innovative Research in Science and Technology*, vol. 1, no. 10, pp. 252–255, 2015.

[25] S. Hoffman, *Apache Flume: Distributed Log Collection for Hadoop*, Packt Publishing Ltd, Maharashtra, India, 2013.

[26] A. Gahlawat, "Big data analytics using R and Hadoop," *International Journal of Computational Engineering and Management*, vol. 1, no. 17, pp. 9–14, 2013.

[27] M. Zaharia, M. Chowdhury, T. Das et al., "Fast and interactive analytics over Hadoop data with Spark," *USENIX Login*, vol. 37, no. 4, pp. 45–51, 2012.

[28] S. Urbanek, M. S. Urbanek, and S. J. JDK, "Package 'rJava'," 2017, http://www.rforge.net/rJava/.

[29] S. Salian and D. G. Harisekaran, "Big data analytics predicting risk of readmissions of diabetic patients," *International Journal of Science and Research*, vol. 4, no. 4, pp. 534–538, 2015.

[30] B. R. Chang, H.-F. Tsai, and C.-M. Chen, "Empirical analysis of cloud-mobile computing based VVoIP with intelligent adaptation," *Journal of Internet Technology*, vol. 17, no. 5, pp. 993–1002, 2016.

[31] Proxmox Virtual Environment, 2017, https://pve.proxmox.com/.

[32] B. R. Chang, H.-F. Tsai, and Y.-C. Tsai, "High-performed virtualization services for in-cloud enterprise resource planning system," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 5, no. 4, pp. 614–624, 2014.

[33] H. Topcuoglu, S. Hariri, and M. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 3, pp. 260–274, 2002.