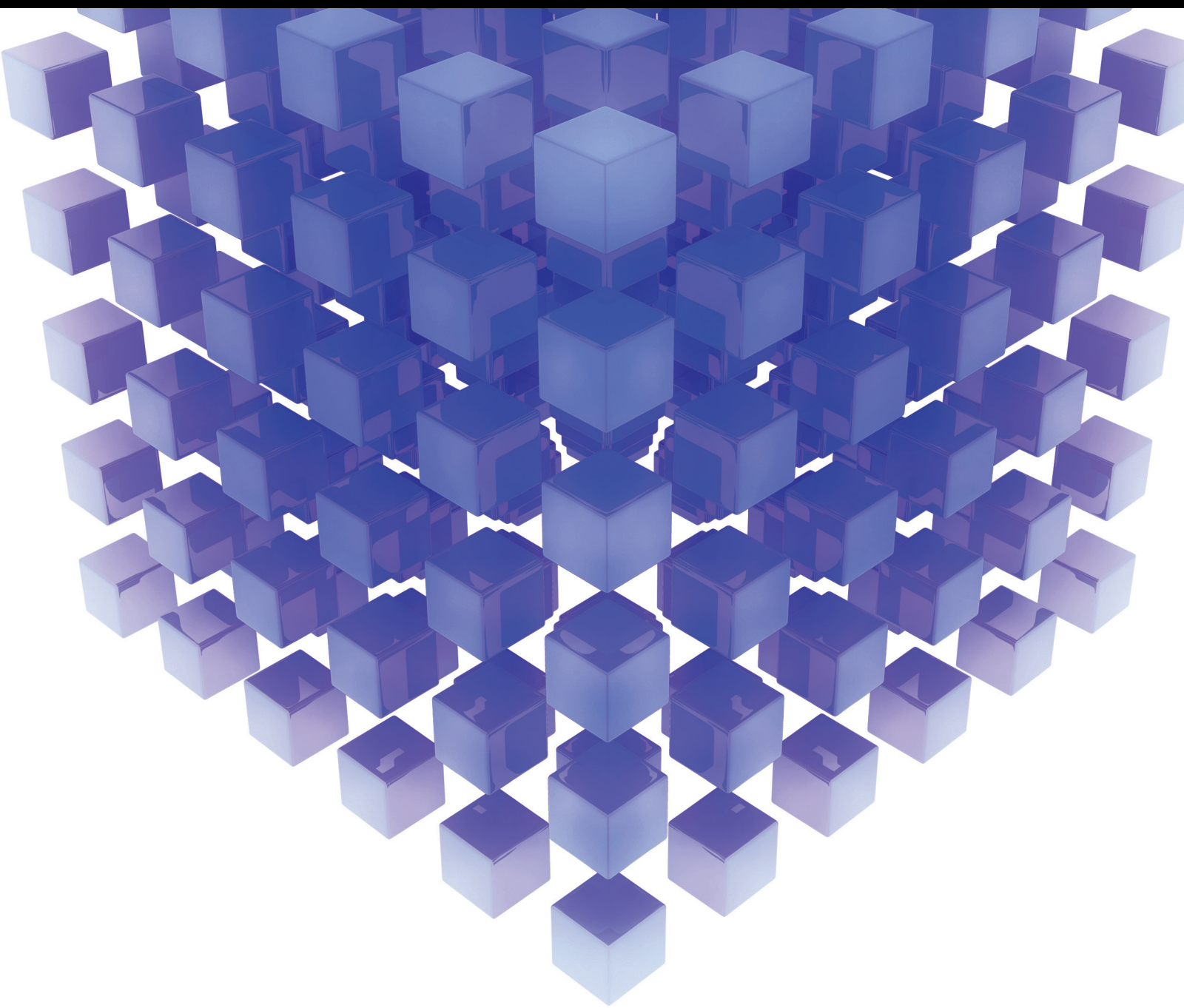
Mathematical Problems in Engineering

Theory and Applications of Data Clustering

Guest Editors: Costas Panagiotakis, Emmanuel Ramasso, Paraskevi Fragopoulou, and Daniel Aloise



Theory and Applications of Data Clustering

Mathematical Problems in Engineering

Theory and Applications of Data Clustering

Guest Editors: Costas Panagiotakis, Emmanuel Ramasso, Paraskevi Fragopoulou, and Daniel Aloise

Editorial Board

Contents

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 5427923, 2 pages
http://dx.doi.org/10.1155/2016/5427923

Editorial

Theory and Applications of Data Clustering

Costas Panagiotakis,[1] Emmanuel Ramasso,[2] Paraskevi Fragopoulou,[3] and Daniel Aloise[4]

[1]*Department of Business Administration, TEI of Crete, 72100 Agios Nikolaos, Greece*
[2]*Department of Automatic Control & Micro-Mechatronic Systems and Applied Mechanics Department, FEMTO-ST Institute, UMR CNRS 6174-UBFC/ENSMM/UTBM, 25000 Besançon, France*
[3]*Department of Informatics Engineering, TEI of Crete, 71004 Heraklion, Greece*
[4]*Department of Computer Engineering and Automation, Federal University of Rio Grande do Norte, 59072-970 Natal, RN, Brazil*

Correspondence should be addressed to Costas Panagiotakis; cpanag@staff.teicrete.gr

Received 17 January 2016; Accepted 17 January 2016

This special issue is particularly focused on fundamental and practical issues in data clustering [1–6]. Data clustering aims at organizing a set of records into a set of groups so that the overall similarity between the records within a group is maximized while minimizing the similarity with the records in the other groups. The data clustering is a state of the art problem with increasing number of applications. For decades, data clustering problems have been identified in many applications and domains such as computer vision and pattern recognition (e.g., video and image analysis for information retrieval, object recognition, image segmentation, and point clustering), networks (e.g., identification of web communities), databases and computing (facing privacy in databases), and statistical physics and mechanics (e.g., understanding phase transitions, vibration control, and fracture identification using acoustic emission data). In addition, several definitions and validation measures [3, 7] of data clustering problem have been used on different applications in engineering. For instance, the goal of the classical clustering problem is to find the clusters that optimize a predefined criterion while the goal of the microaggregation problem [8] is to determine the clusters under the constraint of a given minimum cluster size for masking microdata.

In this special issue, the selected papers focus on the topics of theory and applications of data clustering. They propose new methods that have been successfully applied on several clustering problems including image segmentation [9, 10], time series clustering [4], graph clustering (community detection) [11, 12], and (stock) recommendation systems [13, 14]. Image segmentation is a key step in many image analysis and interpretation tasks. Finding semantic regions is the ultimate goal of segmentation for image understanding. It has become a necessity for many applications, such as content based image retrieval and object recognition. The goal of time series clustering is to partition time series into clusters based on similarity or distance criteria, so that time series in the same group are similar and dissimilar to the time series in the other groups. Concerning the community detection problem, it holds that networks are usually composed of subgroup structures, whose interconnections are sparse and the intraconnections are dense, which is called community structure. Detecting the community structure of a network is a fundamental problem in complex networks which presents many variations. Community detection is often a NP-hard problem and traditional methods for detecting communities in networks can be concluded into two categories: graph partitioning and hierarchical clustering. The recommender system tries to predict the behavior of a complex system by producing a list of recommendations. In stock recommendation that has become a hot topic, most of the methods try to integrate multiple technologies, such as data mining, machine learning, herd psychology, and other nontraditional technologies.

During the last decades, there have been published thousands of clustering algorithms [1]. The clustering methods can be classified into five major categories [2]: partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods. A partitioning method constructs (crisp or fuzzy) partitions of the data, where each partition represents a cluster. The

partition is called crisp if each object belongs to exactly one cluster or fuzzy if one object is allowed to belong to more than one cluster at the same time. Hierarchical clustering algorithms recursively find nested clusters either in agglomerative (bottom-up) mode or in divisive (top-down) mode. Agglomerative algorithms start with each point as a separate cluster and successively merge the most similar pair of clusters. On the contrary, divisive algorithms start with all the data points in one cluster and recursively divide each cluster into smaller clusters. In both cases, a hierarchical structure (e.g., dendrogram) is provided which represents the merging or dividing steps of the method. The density-based methods continue growing a cluster as long as its density (number of data objects in the "neighborhood") exceeds a threshold. Concerning the grid-based methods, they quantize the object space into a finite number of cells that form a grid structure. Then, they use statistical attributes for all the data objects located in each individual cell and clustering is performed on the grid, instead of data objects themselves. Model-based methods assume a model for each of the clusters and attempt to best fit the data to the assumed model.

The definition of a metric that can be used to validate clusters of different densities and/or sizes is an open problem. In the literature, several clustering validity measures have been proposed to measure the quality of clustering [3, 7, 15]. In addition, using the clustering validity measures, it is possible to compare the performance of clustering algorithms and to improve their results by getting a local minima of them.

The papers, published in this special issue, have novelty and contain some interesting methods and applications on data clustering. We believe that the papers published in this special issue will motivate further research in the field of data clustering.

Acknowledgments

The guest editors wish to express their sincere gratitude to the authors and reviewers who contributed greatly to the success of this special issue. We would also like to thank the editorial board members of this journal for their support and help throughout the preparation of this special issue.

Costas Panagiotakis
Emmanuel Ramasso
Paraskevi Fragopoulou
Daniel Aloise

References

[1] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[2] T. W. Liao, "Clustering of time series data—a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[3] C. Panagiotakis, "Point clustering via voting maximization," *Journal of Classification*, vol. 32, no. 2, pp. 212–240, 2015.

[4] E. Ramasso, V. Placet, and M. L. Boubakar, "Unsupervised consensus clustering of acoustic emission time-series for robust damage sequence estimation in composites," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 12, pp. 3297–3307, 2015.

[5] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "NP-hardness of Euclidean sum-of-squares clustering," *Machine Learning*, vol. 75, no. 2, pp. 245–248, 2009.

[6] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, and A. C. P. L. F. de Carvalho, "A survey of evolutionary algorithms for clustering," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 39, no. 2, pp. 133–155, 2009.

[7] C.-H. Chou, M.-C. Su, and E. Lai, "A new cluster validity measure and its application to image compression," *Pattern Analysis and Applications*, vol. 7, no. 2, pp. 205–220, 2004.

[8] C. Panagiotakis and G. Tziritas, "Successive group selection for microaggregation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 5, pp. 1191–1195, 2013.

[9] C. Panagiotakis, H. Papadakis, E. Grinias, N. Komodakis, P. Fragopoulou, and G. Tziritas, "Interactive image segmentation based on synthetic graph coordinates," *Pattern Recognition*, vol. 46, no. 11, pp. 2940–2952, 2013.

[10] C. Panagiotakis, I. Grinias, and G. Tziritas, "Natural image segmentation based on tree equipartition, bayesian flooding and region merging," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2276–2287, 2011.

[11] H. Papadakis, C. Panagiotakis, and P. Fragopoulou, "Distributed detection of communities in complex networks using synthetic coordinates," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2014, no. 3, Article ID P03013, 2014.

[12] D. Aloise, G. Caporossi, P. Hansen, L. Liberti, S. Perron, and M. Ruiz, "Modularity maximization in networks by variable neighborhood search," in *Graph Partitioning and Graph Clustering*, vol. 588, pp. 113–128, American Mathematical Society, 2013.

[13] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[14] E. J. de Fortuny, T. De Smedt, D. Martens, and W. Daelemans, "Evaluating and understanding text-based stock price prediction models," *Information Processing & Management*, vol. 50, no. 2, pp. 426–441, 2014.

[15] M. Sedlmair, A. Tatu, T. Munzner, and M. Tory, "A taxonomy of visual cluster separation factors," *Computer Graphics Forum*, vol. 31, no. 3, part 4, pp. 1335–1344, 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 3790590, 15 pages
http://dx.doi.org/10.1155/2016/3790590

Research Article

A Neighborhood-Impact Based Community Detection Algorithm via Discrete PSO

Dongqing Zhou and Xing Wang

Aeronautics and Astronautics Engineering College, Air Force Engineering University, Xi'an 710038, China

Correspondence should be addressed to Dongqing Zhou; qq_eastz@126.com

Received 30 August 2015; Revised 6 December 2015; Accepted 31 December 2015

Academic Editor: Daniel Aloise

The paper addresses particle swarm optimization (PSO) into community detection problem, and an algorithm based on new label strategy is proposed. In contrast with other label propagation strategies, the main contribution of this paper is to design the definition of the impact of node and take it into use. Special initialization and update approaches based on it are designed in order to make full use of it. Experiments on synthetic and real-life networks show the effectiveness of proposed strategy. Furthermore, this strategy is extended to signed networks, and the corresponding objective function which is called modularity density is modified to be used in signed networks. Experiments on real-life networks also demonstrate that it is an efficacious way to solve community detection problem.

1. Introduction

Complex networks have attracted increasing attention of researchers from different fields, such as physics, sociology, mathematics, and computer science [1]. The related theory has been widely applied in many aspects, including the Internet [2], communication [3], biology [4], and economy [5]. Networks are usually composed of subgroup structures, whose interconnections are dense and intraconnections are sparse. This property is called community structure. Detecting community structure is one of the fundamental issues in networks study; it could reveal latent meaningful structure in networks. It is particularly important to detect the structure of the commonly used networks, such as daily social networks, recommendation system, and nation power distribution networks [6].

Community detection is an NP-hard problem [7]; traditional methods for detecting communities in networks can be concluded in two categories: graph partitioning and hierarchical clustering. The graph partitioning method has been widely used in computer science and the related fields [8]. However, this method needs to know the number of communities and the size of them before partitioning. The hierarchical clustering methods include agglomerative clustering algorithm and divisive clustering algorithm; they do not require the number or size of the communities [9, 10]. However, the results of this method depend on the specific similarity measure adopted.

In recent years, a lot of effort has been made to develop new approaches and algorithms to detect and quantify community structure in complex networks. Newman introduces the modularity as a stopping criterion originally, which becomes one of the most commonly used and best known quality functions. A lot of modularity based methods have been proposed, including modularity optimization, simulated annealing [11], extremal optimization [12], and spectral optimization [13]. These modularity based methods provide an outstanding way to solve the community problem and many researches are carried out based on these methods. Pizzuti models the community problems as single-objective optimization problem and multiobjective optimization problem, respectively, in [14] and [15]. Arenas and Díaz-Guilera investigate the connection between the dynamics of synchronization and the modularity on complex networks in [16].

However, the modularity has the disadvantage of resolution limits found by Fortunato and Barthélemy in [17]. It is that the modularity has an intrinsic scale and the modularity may not be resolved even in the extreme case if it is smaller than this scale.

Li et al. propose a quantitative measure called the modularity density, which uses the average modularity degree to evaluate the partition of a network in [18]. This method provides a mesoscopic way to describe the network structure and overcomes the resolution limit of modularity. The larger the value of modularity density is, the more accurate a partition is. Thus, the community detection can be viewed as an optimization problem of finding a partition of a network which has the maximum modularity density.

Particle swarm optimization (PSO) algorithm is successfully used in many optimization fields [19–22]. It was proposed to solve continuous optimization problems at first; then Kennedy and Eberhart develop the continuous PSO to discrete binary PSO in [23]. Chen et al. propose the candidate solution and velocity based on discrete PSO in [24]. Recently, Gong et al. use discrete PSO to solve the community detection problem in [25]. Compared with traditional optimization methods, the discrete PSO is simple to implement and it has a high speed of convergence. It needs to know neither the number of the communities nor the size of the communities. What is more, bare mathematical assumptions that may be needed in the conventional methods are ignored. These advantages make it feasible and promising in solving community detection problems.

In this paper, a new algorithm based on discrete PSO is proposed to solve community detection problems by optimizing the modularity density function. We design a definition called the impact of node in this paper, which takes the neighborhood information into consideration. A new initialization and updating strategy based on the impact of node is proposed. At last, we modify the modularity density function to signed networks according to the character of community, and the proposed strategies are also extended to signed networks.

The rest of this paper is organized as follows. Section 2 is a review of the related introduction of modularity density. Section 3 gives a detailed description of the proposed method. In Section 4, the experimental results of the proposed algorithm in comparison with other approaches are shown. At last the conclusions are summarized in Section 5.

2. Community Structure and Modularity Density

A network can be modeled as $G = (V, E)$, where V and E represent the vertices and edges, respectively. Assume that $\mathbf{A}$ is the adjacency matrix of G. If there is a link between nodes i and j, $A_{ij} = 1$; otherwise $A_{ij} = 0$. Suppose that S is subgraph which belongs to G, and i is a node which belongs to S. k_i is the degree of the node i; $k_i^{\text{in}} = \sum_{i,j\in S} A_{ij}$, $k_i^{\text{out}} = \sum_{i\in,j\notin S} A_{ij}$. The community of a network usually has the following property:

$$\sum_{i\in S} k_i^{\text{in}} > \sum_{i\in S} k_i^{\text{out}}. \tag{1}$$

It means that the sum of all degrees within the community is larger than the sum of all degrees toward the rest of the network.

If the links between different nodes have negative or positive signs, they are called the signed networks. The signed networks can be modeled as $G = (V, \text{PE}, \text{NE})$. V is the set of nodes; PE and NE are the positive and negative links, respectively. Let e_{ij} be the link between nodes i and j; then the adjacency matrix $\mathbf{A}$ can be formulated as follows: if $e_{ij} \in \text{PE}$, $A_{ij} = 1$; if $e_{ij} \in \text{NE}$, $A_{ij} = -1$. Then the signed networks usually have the following property:

$$\begin{aligned} \sum_{i\in S} \left(k_i^{+}\right)^{\text{in}} &> \sum_{i\in S} \left(k_i^{+}\right)^{\text{out}}, \\ \sum_{i\in S} \left(k_i^{-}\right)^{\text{in}} &< \sum_{i\in S} \left(k_i^{-}\right)^{\text{out}} \end{aligned} \tag{2}$$

in which

$$\begin{aligned} \left(k_i^{+}\right)^{\text{in}} &= \sum_{i,j\in S, e_{ij}\in \text{PE}} A_{ij}, \\ \left(k_i^{-}\right)^{\text{in}} &= \sum_{i,j\in S, e_{ij}\in \text{NE}} \left|A_{ij}\right|, \\ \left(k_i^{+}\right)^{\text{out}} &= \sum_{j\in S, e_{ij}\in \text{PE}} A_{ij}, \\ \left(k_i^{-}\right)^{\text{out}} &= \sum_{j\in S, e_{ij}\in \text{NE}} \left|A_{ij}\right|. \end{aligned} \tag{3}$$

In an intuitive way, take the real-life networks SPP and GGS as examples (please refer to Section 4.5) for signed networks; the solid line and dashed line represent positive and negative links, respectively. The internal positive degree in a community is dense, and the negative external degree between different communities is dense.

Community detection can be formulated as a *modularity* (Q) optimization problem. Q was proposed by Newman in [13], which aims at finding a partition of the network. Suppose that there is a graph G' whose edges are drawn at random; it has the same distribution of degrees as G. Modularity is such a measurement that maximizes the sum of the inner edges over all the modules of G minus that of G' which has the expected sum of number of inner edges [26]. As Q is an evaluation function to estimate the community structure of the network, the larger the value of Q is, the clearer the structure of the community is. Otherwise, the structure is more obscure. The mathematical description of Q is as follows:

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(i, j), \tag{4}$$

where m is the number of edges in the network. If i and j are in the same community, $\delta(i, j) = 1$. Otherwise, $\delta(i, j) = 0$. According to [18], another form of Q is as follows:

$$Q = \sum_{i=1}^{N} \left[\frac{L(V_i, V_i)}{L(V, V)} - \left(\frac{L(V_i, V)}{L(V, V)} \right)^2 \right], \tag{5}$$

where N is the number of communities and $L(V_k, V_n) = \sum_{i \in V_k, j \in V_n} A_{ij}$, $L(V, V) = 2m$, and $L(V_i, V) = \sum_{j \in V_i, k \in V} A_{jk}$.

A class of methods aiming to maximize the modularity has been developed. However, the modularity has the disadvantage of resolution limits that it contains an intrinsic scale which depends on the size of links in the network. It cannot detect them exactly when the modules are smaller than this scale.

Modularity density (D) was proposed by Li et al. in [18] to evaluate the partition of a network based on the concept of average modularity degree. It overcomes the resolution limit in community detection:

$$D = \sum_{i=1}^{N} \frac{L\left(V_i, V_i\right) - L\left(V_i, \overline{V_i}\right)}{\left|V_i\right|}. \tag{6}$$

$L(V_i, V_i)/|V_i|$ and $L(V_i, \overline{V_i})/|V_i|$ mean the average internal and external degrees of the ith community, respectively. D tries to maximize the average internal degree and minimize the average external degree of the communities. D is related to the density of subgraphs; it provides a way to overcome the problem that Q is sensitive to the the size of network and interconnections of modules. Thus, we could use D to decide whether the networks are partitioned into correct communities. According to the definition of modularity density D, the larger the value of D is, the more accurate a partition is. Then Li et al. improved D to a general version by setting a parameter λ to the proportion of average internal degree and external degree:

$$D_\lambda = \sum_{i=1}^{N} \frac{2\lambda L\left(V_i, V_i\right) - 2\left(1-\lambda\right) L\left(V_i, \overline{V_i}\right)}{\left|V_i\right|}. \tag{7}$$

D_λ is a convex combination of ratio cut and ratio association. It tries to maximize the density of links inside a community and minimize the density of links among different communities. When $\lambda = 1$, D_λ is equal to ratio association; when $\lambda = 0$, D_λ is equal to ratio cut; when $\lambda = 0.5$, D_λ is equal to D. We can decompose the network into large communities when using small λ; otherwise, small communities are obtained. And for this, more details and levels of the network can be found. In this paper we use the discrete PSO algorithm to optimize D_λ to detect community structure.

3. The Description of Proposed Algorithm

In this section, the proposed algorithm for community detection is described in detail. First, the objective function of the algorithm is given. Next, the impact of node is described. Meanwhile, the particle swarm initialization and updating are presented. At last, the framework of the proposed algorithm is elaborated, and the complexity analysis is presented.

3.1. Objective Function. In this paper, we adopt D_λ as the objective function because of its efficiency in detecting community structure, which has been described in detail in Section 2. In order to solve signed network problem, we extend it to (8). It is consistent with the character of signed networks:

$$\begin{aligned} D_\lambda^+ &= \sum_{i=1}^{N} \frac{2\lambda L\left(V_i, V_i\right)^+ - 2\left(1-\lambda\right) L^+\left(V_i, \overline{V_i}\right)}{\left|V_i\right|}, \\ D_\lambda^- &= -\sum_{i=1}^{N} \frac{2\lambda L\left(V_i, V_i\right)^- - 2\left(1-\lambda\right) L^-\left(V_i, \overline{V_i}\right)}{\left|V_i\right|}, \\ D_\lambda &= D_\lambda^+ + D_\lambda^-. \end{aligned} \tag{8}$$

3.2. The Impact of Node. In order to solve community detection problem with discrete PSO, label propagation is introduced in [25]. The label of node is used to assign the node to different communities, and the nodes which have the same label are considered to be in the same community. This label propagation strategy considers the number of nodes with the same label in the neighborhood to update the current node's label. For example, consider a node i whose neighbors are $i^1, i^2, \ldots, i^n$, when using this label propagation to update the label of node i, check all the labels of $i^1, i^2, \ldots, i^n$ to find the label which appears the most, and then node i is assigned to this label. But actually, the contribution of each node to the neighborhood is not the same as the node we choose. In this subsection, a new definition that defines the "impact of node" is introduced and a new label propagation based on the "impact of node" is proposed. Consider a node i whose neighbors are $i^1, i^2, \ldots, i^k$; then the impact of node i^j on node i can be defined as follows:

$$\text{ion}\left(i^j\right) = \text{degree}\left(i^j\right) * \text{num}\left(l\left(i^j\right)\right), \tag{9}$$

where ion is the impact of node, degree (i^j) is the degree of node i^j, $l(i^j)$ means the label of node i^j, and num $(l(i^j))$ refers to the number of neighbors of i that have the same label with node i^j. It is noticed that the impact of node is usually different when the node we choose is in the neighborhood of different nodes. For signed networks, we only consider the positive links for that the number of negative links in a community is usually less than that of negative links between communities, and two nodes connected with negative links are usually located in different communities.

The ion describes the effectiveness of the node in the network, and the value of the ion can show the connection level in the corresponding local network. The bigger the ion is, the tighter the connection will be. So, the ion can be used as a measure to detect which community the node belongs to.

Suppose that the neighborhood of node i has the label set $\Omega(i) = (l(i^1), l(i^2), \ldots, l(i^k))$; the new label propagation based on the impact of node is as follows:

$$\text{LP}\left(i\right) = \underset{l(i^j) \in \Omega(i)}{\text{argmax}} \left(\text{ion}\left(i^j\right)\right). \tag{10}$$

It means that the label of current node is decided by its neighbour which has the biggest impact of node.

We choose the karate network to illustrate the rationality of the impact of node. The karate network was completed by

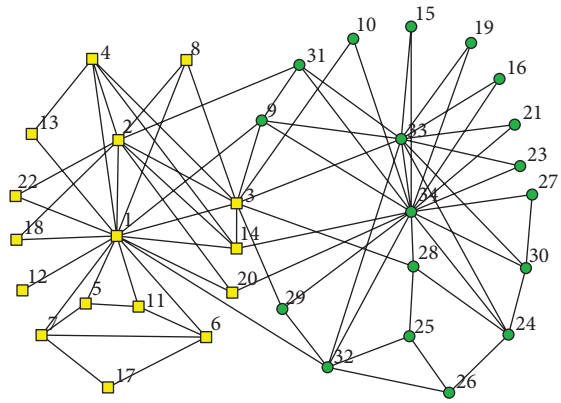


Figure 1: The real community structure of karate network.

Gong et al. after two years' observation. The club splits into two groups because of the dispute between the administrator and the instructor of the club. Figure 1 shows the real community structure of karate network; the nodes with the same mark belong to the same community. Vertices 1 and 34 represent the administrator and the instructor, respectively. When deciding the labels of each node in the community, there are three cases needed to consider. The first case is that when all the neighbors of a node have different labels, then the degree of node will decide the label of the current node; the other case is that when some of the neighbors of the current node share the same label, then the impact of these neighbors on the current node depends on the number of neighbors that share the same label if they have similar degree; otherwise, both of the two factors (degree (i^j) and num $(l(i^j))$) will play important role in deciding the label of the current node. Take node 3 as an example; its neighbors are $\{1, 2, 4, 8, 14, 9, 10, 28, 29, 33\}$, and their degrees are $\{16, 9, 5, 3, 5, 5, 2, 4, 3, 12\}$. If all the labels of these neighbors are different, node 3 will be assigned to the same label with node 1 since the ion of node 1 is the biggest according to our proposed strategy. If all the neighbors of node 3 have been assigned to the right communities, the ion of all these neighbors to node 3 is $\{80, 45, 25, 45, 25, 25, 10, 20, 15, 60\}$; then node 1 has the highest impact on node 3, and node 3 is assigned to the same community with node 1. However, we can not decide the label of node 3 if we use label propagation method in [25] in both cases.

3.3. Particle Swarm Initialization. In PSO algorithm, proper initialization scheme can generate a set of high quality solutions and reduce the searching time significantly. Traditional initialization method generates solutions randomly. It does not take the adjacency matrix of the community into consideration, while this matrix usually provides important information to the optimization. In order to make use of this prior knowledge, we put forward a new particle swarm initialization based on the impact of node in this paper. The procedure is as follows.

Step 1. Initialize every node with unique labels, $l(i) = i$.

Step 2. Sort all the nodes in descending order according to their degree.

Step 3. Initialize the ith particle, select the node which has the ith highest degree (denoted as node a), and find the node (denoted as b) which has the maximal degree connected with node a. Assign all the nodes connected with both node a and node b to the same label.

Step 4. Repeat Step 3 until all the networks are gone through or the termination criteria are met.

If the number of nodes is less than particles, select two linked nodes randomly and assign them to the same label for the rest particles. For signed networks, only the positive links are considered in Steps 3 and 4.

The description of proposed initialization method can be seen in Figure 2. According to the adjacency graph, node 6 has the greatest degree among all the nodes, so it is selected firstly. The degree of node 10 is the biggest in all the neighbors of node 6. In this way, nodes (6, 7, 8, 9, 10, 11) will be assigned to the same label. But if we simply assign the nodes connected to node 6 to the same label based on label propagation in [25], the nodes (4, 5, 6, 7, 8, 9, 10, 11) will share the same label. Actually, node 4 and node 5 should have the same label with node 3 according to the structure of the network. The advantage of proposed initialization scheme can be seen from this example.

3.4. Particle Status Updating. Particle status updating is a key process in PSO algorithm. It consists of velocity updating and position updating. In this paper, velocity updating uses the update rule in [25], and the position updating depends on the velocity updating and the proposed impact of node. The velocity updating rule in discrete form is as follows:

$$V_i = \text{sig}\left(\omega V_i + c_1 r_1 \left(pbest \oplus X_i\right) + c_2 r_2 \left(gbest \oplus X_i\right)\right), \quad (11)$$

where ω is the inertia weight and c_1 and c_2 are the cognitive and social components, respectively. r_1 and r_2 are two random numbers which range from 0 to 1. In this paper, the inertia weight ω is randomly generated within $[0, 1]$, and the cognitive and social components c_1 and c_2 are set to the typical value of 1.494 [25]. $\oplus$ is defined as an XOR operator. Suppose that $Y = (y_1, y_2, \ldots, y_n)$ and $X = (x_1, x_2, \ldots, x_n)$; the function $Y = \text{sig}(X)$ is defined as follows:

$$y_i = \begin{cases} 1, & \text{rand}\,(0,1) < \text{sigmoid}\,(x_i) \\ 0, & \text{rand}\,(0,1) \geq \text{sigmoid}\,(x_i). \end{cases} \quad (12)$$

The sigmoid function is defined as

$$\text{sigmoid}\,(x) = \frac{1}{1 + e^{-x}}. \quad (13)$$

Now the position updating rule is defined as the following discrete form:

$$X_{i+1} = X_i \otimes V_i, \quad (14)$$

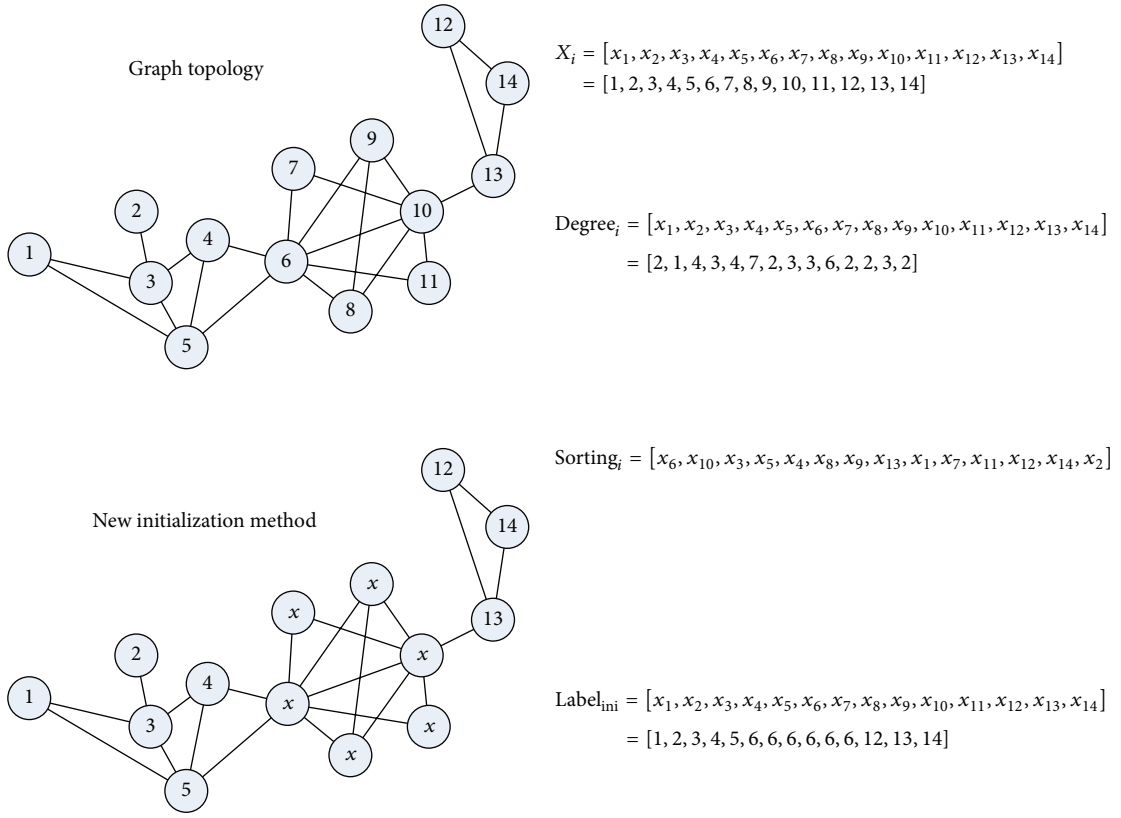


Figure 2: Description of proposed initialization method.

where the new position is generated by "position ⊗ velocity," that is, given a position $X_1 = (x_{11}, x_{12}, \ldots, x_{1n})$ and a velocity $V = (v_1, v_2, \ldots, v_n)$. The element of new position $X_2 = (x_{21}, x_{22}, \ldots, x_{2n})$ is defined as

$$x_{2i} = \begin{cases} x_{1i}, & v_i = 0 \\ Nbest_i, & v_i = 1, \end{cases} \tag{15}$$

where $Nbest_i$ is calculated by new label propagation in (10) which is based on the impact of node:

$$Nbest_i = \underset{l(i^j)\in\Omega(i)}{\operatorname{argmax}} \left(\operatorname{ion}\left(i^j\right)\right). \tag{16}$$

The procedure of the new label propagation can be described as follows:

(1) Count the labels of all the nodes connected with the current node, which are called neighborhood nodes in the following steps.

(2) Calculate the impact of these nodes according to (9).

(3) Find the label of node which has the greatest ion and assign this label to the current node; if there are more than 2 labels which have the same votes, select one label randomly to assign it to the current node.

It is noted that this update strategy is more effective in the later stage of community detection because we often encounter the situation that a node's neighbors belong to different communities. However, we do not know when the algorithm converges to a stable state on different networks. So, we use the update strategy in [25] in uneven number generation because of its simplicity, and we use the proposed update strategy in even number generation because of its effectivity.

3.5. Framework of the Proposed Algorithm. After illustrating all the corresponding strategies in detail, the framework of the proposed algorithm is presented in Algorithm 1.

3.6. Complexity Analysis. The main complexity of the proposed algorithm relies on the cycling process. Suppose that the network we tested has n nodes and m edges. In Algorithm 1, Step 1 can be finished in linear time. The time complexity of Steps 2.1 and 2.2 is $O(n)$ and $O(2m)$ in the worst case, respectively. Step 2.3 can be accomplished in $O(m + n)$ time; Steps 2.4 and 2.5 need $O(1)$ operation. So, the total time complexity of the proposed algorithm is $O(\text{pop} \times \text{maxgen} \times \max\{(m+n), (2m), O(n)\})$, which can be simplified as $O(\text{pop} \times \text{maxgen} \times (2m))$.

4. Experiment Discussion

In this section, detailed experiments are carried out to test the effectiveness of the proposed algorithm against other five algorithms or methods, including CNM, FTQ, Informap,

(1) **Input**:
Adjacency matrix A;
Population size: pop;
Maximum generation: maxgen;
Tuning parameter: λ;
(2) **Step 1**: Initialization
(1.1) Position initialization: $\mathbf{x} = (x_1, x_2, \ldots, x_{\text{pop}})^T$;
(1.2) Velocity initialization: $\mathbf{v} = (v_1, v_2, \ldots, v_{\text{pop}})^T$;
(1.3) Personal best position initialization: **pbest** $= (pbest_1, pbest_2, \ldots, pbest_{\text{pop}})^T$, **pbest** $= \mathbf{x}$;
(1.4) Global best position initialization: select the global best position **gbest** in **pbest**, **gbest** $= (gbest_1, gbest_2, \ldots, gbest_{\text{pop}})^T$.
(3) **Step 2**: Iteration
(2.1) Calculate new velocity according to (11), $\mathbf{v}^{t+1} = (v_1^{t+1}, v_2^{t+1}, \ldots, v_{\text{pop}}^{t+1})^T$;
(2.2) Calculate new position according to (14), (15), and (16)
$\mathbf{x}^{t+1} = (x_1^{t+1}, x_2^{t+1}, \ldots, x_{\text{pop}}^{t+1})^T$;
if $\text{mod}(t, 2) = 0$
Proposed strategy is adopted
else
Strategy in [25] is adopted
endif
(2.3) Function evaluation;
(2.4) Update personal best position **pbest**;
(2.5) Update global best position **gbest**.
(4) **Step 3**: Termination Criteria
if maxgen is arrived
Stop and output **gbest**;
else
Go back to **Step 2**;
end if
(5) **Output**: *gbest*.

Algorithm 1: Framework of the proposed algorithm.

and other two nature inspired algorithms GA-net and label propagation in Gong et al.'s literature [25].

CNM was presented by Clauset et al. in [10]. This method is essentially a fast implementation of the GN approach. GA-net is a single-objective algorithm proposed by Pizzuti in [14]. The algorithm optimizes a simple but efficacious fitness function, which is called community score, to identify densely connected groups. Fine-tuned modularity density algorithm (denoted as FTQ) is a fine-tuned algorithm based on modularity density proposed by Chen et al.; it detects the community structure via splitting and merging the network [6]. Informap is introduced by Rosvall and Bergstrom in [27]. This algorithm uses a new information theoretic approach to reveal community structure in weighted and directed networks. In [25], a multiobjective algorithm based on MOEA/D and PSO is combined to detect the structure of community. Considering the objective function we adopted is a single-objective function, only the label propagation and particles updating strategy in [25] are used and combined with modularity density function (denoted as DPSO in the following) to compare with proposed algorithm.

4.1. Experimental Settings. In this paper, we choose *modularity* Q and *Normalized Mutual Information* (NMI) as the measurement when the ground truth of a network is known. Otherwise, only modularity Q is adopted. For signed networks, SQ is adopted to evaluate the performance in [28], which is formulized as

$$\text{SQ} = \frac{1}{2\omega^+ + 2\omega^-} \sum_{i,j} \left(\omega_{ij} - \left(\frac{\omega_i^+ \omega_j^+}{2\omega^+} - \frac{\omega_i^- \omega_j^-}{2\omega^-} \right) \right) \delta(i, j), \tag{17}$$

where ω_i^+ and ω_i^- represent the sum of all positive and negative weights of node i, respectively, and ω_{ij} represents the weight of adjacency matrix of the signed network.

The *Normalized Mutual Information* (NMI) described in [29] is an index to estimate the similarity between two community detection results. If we assume that A, B are the two partitions of a network and C is the confusion matrix, then $\text{NMI} \in [0, 1]$ measures the similarity between A and B. The larger the value of NMI is, the more similar the partitions A and B are. If $A = B$, $\text{NMI}(A, B) = 1$; if A and B are completely different, $\text{NMI}(A, B) = 0$. $\text{NMI}(A, B)$ can be calculated as follows:

$$\text{NMI}(A, B) = \frac{-2 \sum_{i=1}^{C_A} \sum_{j=1}^{C_B} C_{ij} \log\left(C_{ij} N / C_{i.} C_{.j}\right)}{\sum_{i=1}^{C_A} C_{i.} \log\left(C_{i.}/N\right) + \sum_{j=1}^{C_B} C_{.j} \log\left(C_{.j}/N\right)}, \tag{18}$$

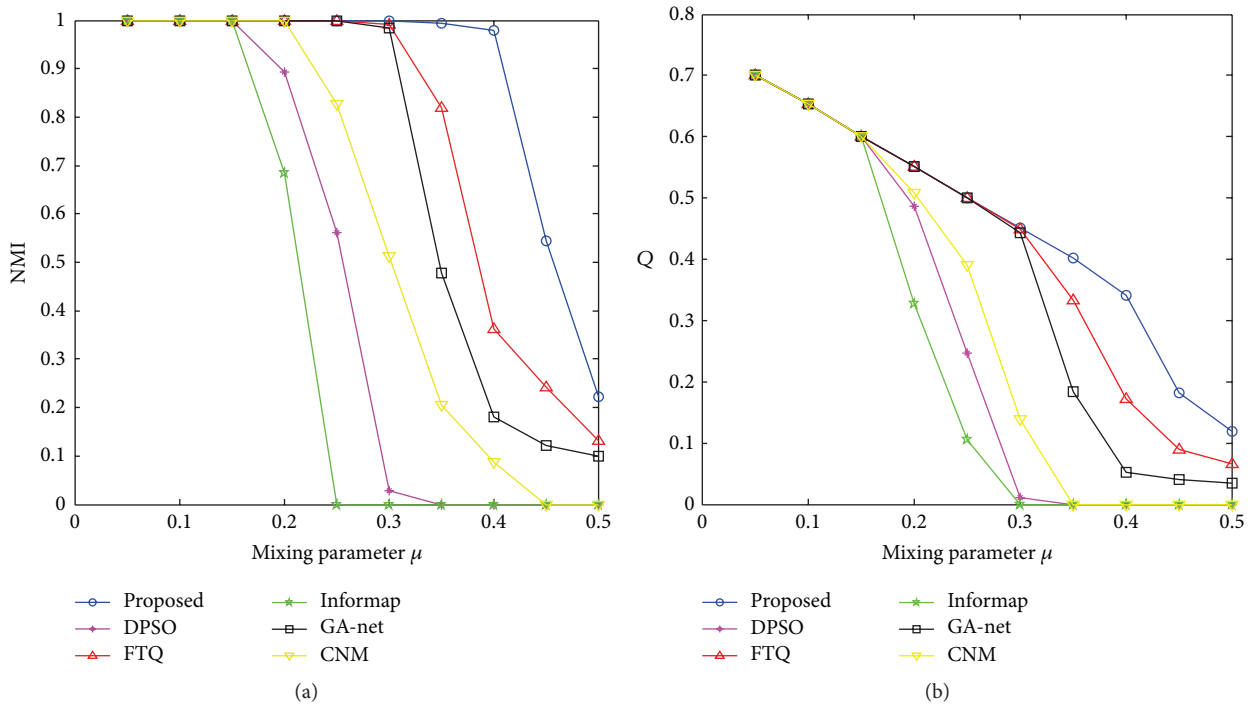


Figure 3: Average maximum NMI and Q values over 30 runs on GN extended benchmark networks.

where C_A (C_B) is the number of the communities in partition A (B) and $C_{i.}$ ($C_{.j}$) is the sum of the elements of C in row i (column j).

In this paper, parameter λ in objective function increases from 0.3 to 0.8 with interval 0.1, and parameter γ in GA-net ranges from 1 to 1.5 with interval 0.1. For each value of parameter λ or γ, all the algorithms run 30 independent times on the test problems with setting pop to 100 and maxgen to 100. Among all the results for each network, the best one is selected and shown in the following experiments.

4.2. Experiments on GN Extended Benchmark Networks. GN extended benchmark network was proposed by Lancichinetti et al. in [30], which is an extension of the classical GN benchmark proposed by Girvan and Newman in [4]. GN extended benchmark network consists of 128 nodes divided into four communities, and each community has 32 nodes. The average degree of node is 16, and mixing parameter μ decides the percentage of connections between communities to the total connections. When $\mu < 0.5$, the network has strong community structure. On the contrary, when $\mu > 0.5$, the community structure is vague, and it is hard to detect its structure.

Experiments on GN extended benchmark networks are done to test the performance of our algorithm. CNM, FTQ, GA-net, Informap, DPSO, and the proposed algorithms are tested on 10 GN extended networks with mixing parameter μ ranging from 0.05 to 0.5. Figure 3 shows the average maximum NMI and Q values obtained from different algorithms when the mixing parameter μ increases from 0.05 to 0.5 with interval 0.05.

As is shown in Figure 3(a), when the mixing parameter $\mu \leq 0.15$, Informap, FTQ, CNM, GA-net, DPSO, and the proposed algorithm can figure out the true partition (NMI equals 1). With the mixing parameter increasing, the community structure of the network becomes fuzzy and it becomes difficult to detect the true structure of the community. Informap and DPSO first show their weakness, and their detection ability decreases rapidly from $\mu = 0.15$ to 0.3. Then detection ability of CNM decreases from $\mu = 0.2$. When $\mu > 0.3$, GA-net and FTQ show its limitation in detecting the community structure. Compared with them, the proposed algorithm shows its superiority. As seen from Figure 3(b), the same conclusion could be derived from measurement Q.

More experiments are discussed on GN extended benchmark network in detail to illustrate the performance of proposed algorithm. In our objective function, λ is a tuning parameter. The bigger the λ is, the more the number of the communities will be detected generally.

Figure 4 shows how NMI changes with mixing parameter when λ adopts different values. As seen from them, DPSO almost gets the same NMI values when λ ranges from 0.3 to 0.8. On the contrary, the proposed algorithm shows its excellent detection ability when community structure gets more and more obscure with the changing of parameter λ. From these figures, the superiority of proposed algorithm is demonstrated. In our view, the designed definition takes the topology of community structure into consideration, which allows the algorithm to detect more obscure structure than the others with a suitable tuning parameter value in the objective functions.

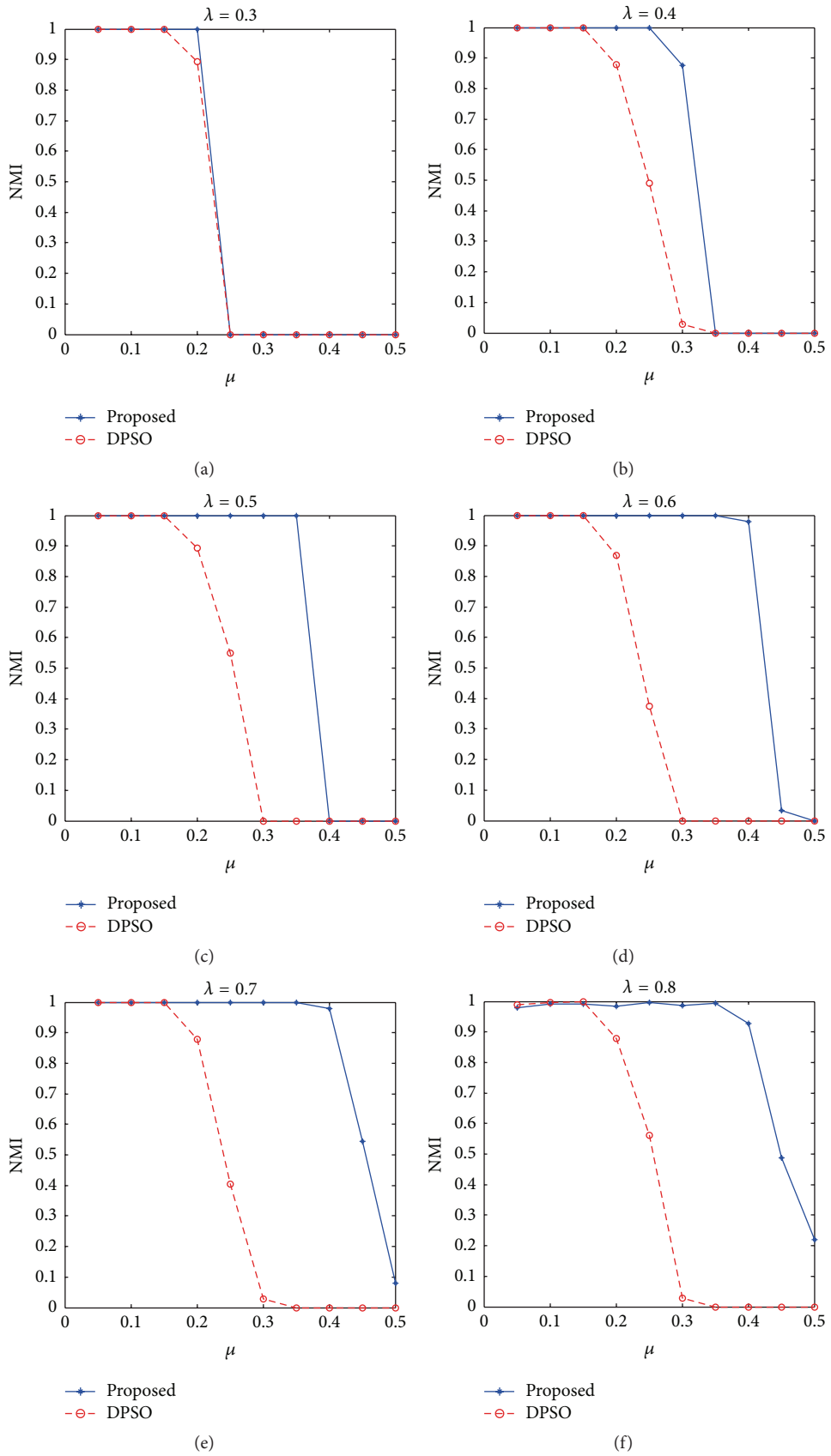
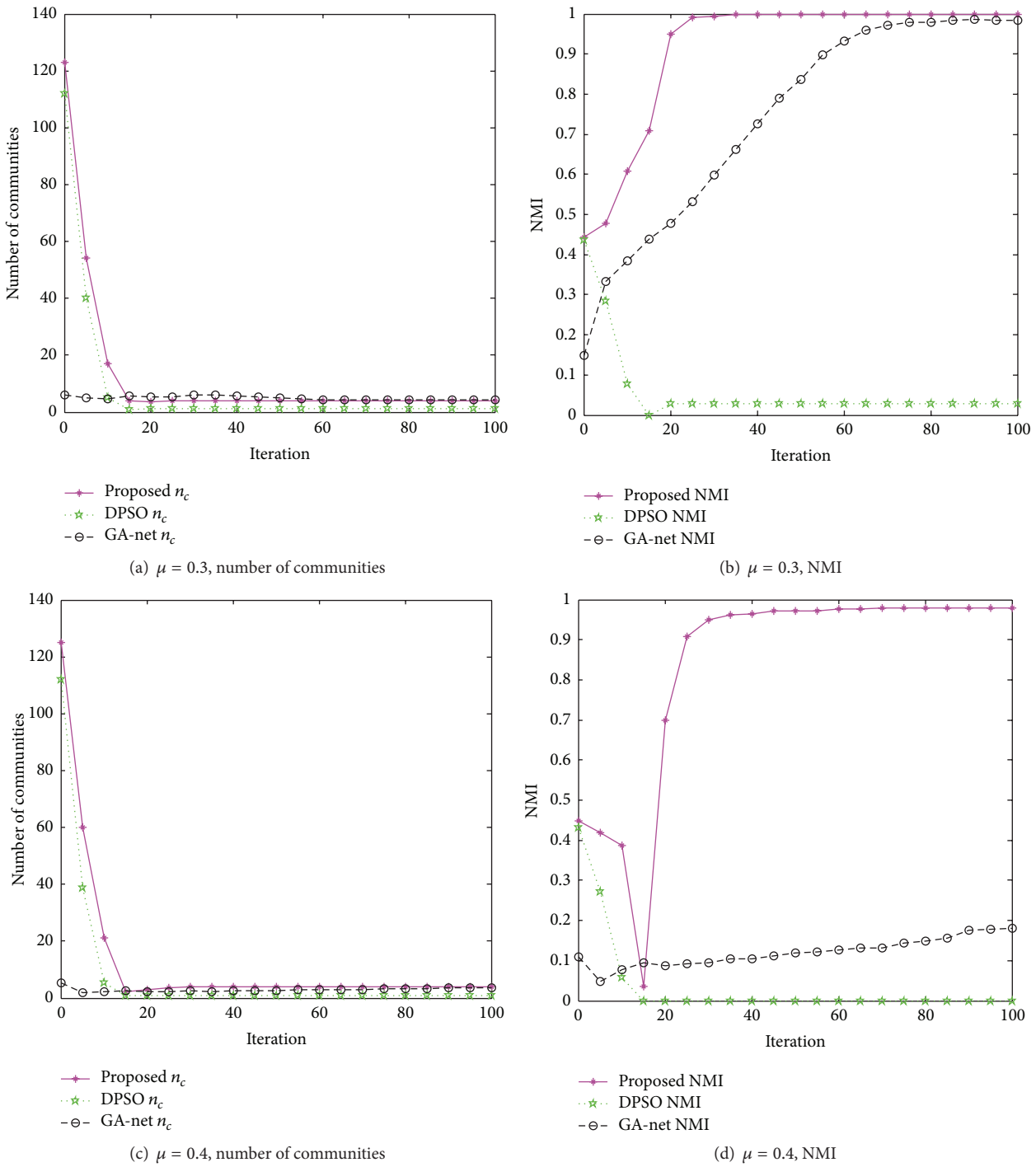


Figure 4: Average maximum NMI values over 30 runs on GN extended benchmark networks with different λ values.



(a) $\mu = 0.3$, number of communities

(b) $\mu = 0.3$, NMI

(c) $\mu = 0.4$, number of communities

(d) $\mu = 0.4$, NMI

Figure 5: Convergence of three algorithms.

In order to discuss the convergence of the nature inspired algorithms (the proposed algorithm, DPSO, and GA-net algorithm), we choose GN extended benchmark networks $\mu = 0.3$ and $\mu = 0.4$ to illustrate it. As shown in Figures 5(a) and 5(c), the number of communities (denoted as n_c in the figures) obtained from all three algorithms converges to a stable value no matter $\mu = 0.3$ or 0.4. During the optimization process, the number of communities has converged to a stable value since the 20th iteration. At the same time, the labels of nodes still keep changing to obtain better results. At last, NMI converges to 1 or almost 1 in about the 30th iteration in Figures 5(b) and 5(d). For DPSO, NMI values are quite low (about equal to 0) since it can not detect the exact number of communities. For GA-net, when $\mu = 0.3$, it can obtain a satisfying result, but it has a slow convergence speed. When the mixing parameter increases to 0.4, it fails

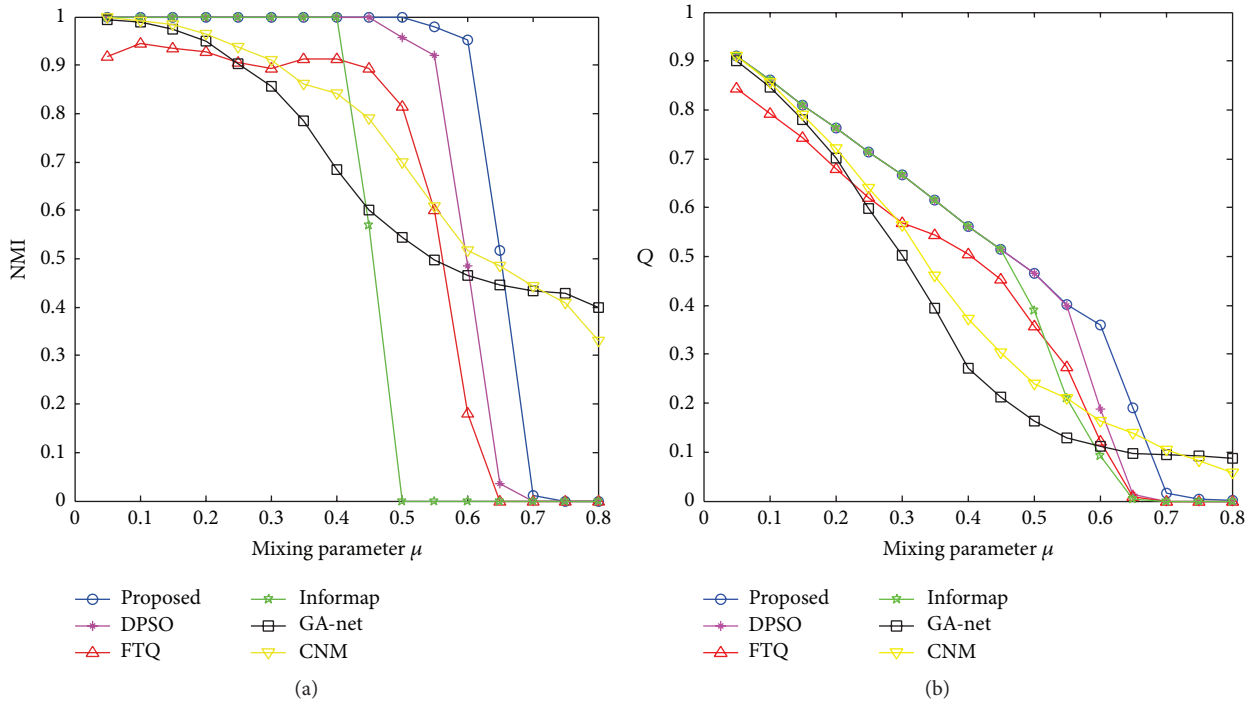


Figure 6: Average maximum NMI and Q values over 30 runs on LFR benchmark networks.

to detect community structure exactly. From Figure 5, a conclusion can be derived that the proposed algorithm significantly outperforms other algorithms and can detect the real structure of a network effectively.

4.3. Experiments on LFR Benchmark Network. On the GN extended benchmark network, all communities have exactly the same size, and the size of network is small, so it cannot reflect some important features in real networks. Because of this, LFR benchmark networks are proposed by Lancichinetti et al. in [30], in which the parameters α and β are set to tune the distribution of degree and community size. Each node shares a fraction $1 - \mu$ of its links with the other nodes in the same community and a fraction μ with the nodes in other communities.

In this paper, we use 16 LFR benchmark networks; their mixing parameter increases from 0.05 to 0.8 with interval 0.05. Each of them consists of 1000 nodes and the cluster size ranges from 10 to 50, $\alpha = 2$ and $\beta = 1$. The averaged degree for each node is 20 and the maximum node degree is 50.

As is shown in Figure 6, the proposed algorithm and DPSO have a better performance than the other 4 algorithms on LFR networks because the proposed algorithm and DPSO share the same objective function. What is more, the detection ability of the proposed algorithm is stronger than that of DPSO with the mixing parameter μ increasing. A conclusion that our initialization and updating schemes can reveal the structure of community more exactly than label propagation in DPSO can be obtained from the comparison result. The performance of GA-net is the worst in all the 6 algorithms when $\mu \leq 0.65$. With the increasing of mixing parameter, GA-net has the best performance when $\mu \geq 0.65$, for the reason that it uses different encoding scheme. DPSO and our proposed algorithm tend to consider all the nodes as a whole community since the structure is too obscure. The experiment on LFR networks also shows the excellent detection ability of the proposed algorithm.

Table 1: Characteristics of adopted real-life networks.

Network	Nodes	Edges	Real communities
Karate	34	78	2
Dolphin	62	159	2
Football	115	616	12
Politics	105	613	3
SFI	118	200	Unknown
Netscience	1589	2742	Unknown

4.4. Experiments on Real-Life Networks. We apply our algorithm to six well-known real-life networks, including Zacharys Karate Club network [31], Dolphin social network [32], American College Football network, Krebs Books on US Politics network, Santa Fe Institute (SFI) network [4], and Netscience network [33]. The characteristics of the networks are shown in Table 1.

Comparison experiments on six real-life networks are tested and shown in Table 2. From what is recorded on the Zacharys Karate Club, Dolphin, and American College Football networks, the proposed algorithm shows a better performance than the other five algorithms. Referring to the Krebs' Books on US Politics network, the performance of the

Table 2: Experiment result on real-life networks.

Dataset	Measurement	Proposed	DPSO	GA-net	FTQ	CNM	Informap
Karate	NMI_{max}	**1.000**	1.000	0.6369	0.7467	0.7154	0.7198
	NMI_{avg}	**1.000**	0.7369	0.6369	0.6956	0.7013	0.7142
	Q_{max}	0.4037	**0.4329**	0.4060	0.3913	0.4095	0.4291
	Q_{avg}	0.3835	**0.4132**	0.4060	0.3796	0.3916	0.4031
Dolphin	NMI_{max}	**1.000**	1.000	0.4236	0.8794	0.6049	0.6395
	NMI_{avg}	**0.9610**	0.9477	0.4060	0.8576	0.5893	0.6043
	Q_{max}	**0.5178**	0.5136	0.4634	0.4914	0.4018	0.4839
	Q_{avg}	**0.5114**	0.5109	0.4634	0.4837	0.3945	0.4839
Politics	NMI_{max}	0.5916	0.5745	0.4403	**0.6792**	0.5701	0.6216
	NMI_{avg}	0.5763	0.5745	0.4201	**0.6513**	0.5620	0.6165
	Q_{max}	0.5036	0.5256	0.5035	**0.5805**	0.5513	0.5674
	Q_{avg}	0.4969	0.5235	0.4798	**0.5604**	0.5470	0.5364
Football	NMI_{max}	**0.9253**	0.9252	0.9252	0.9013	0.7438	0.6574
	NMI_{avg}	**0.9141**	0.9094	0.9107	0.8984	0.7438	0.6574
	Q_{max}	**0.6093**	0.6037	0.6012	0.4892	0.5394	0.3484
	Q_{avg}	**0.6021**	0.6015	0.5906	0.4519	0.5394	0.3484
SFI	Q_{max}	0.7325	**0.7489**	0.5716	0.6946	0.4865	0.5319
	Q_{avg}	**0.7287**	0.7252	0.5620	0.6743	0.4617	0.5319
Netscience	Q_{max}	**0.9474**	0.9206	0.8739	0.9031	0.8021	0.7183
	Q_{avg}	**0.9400**	0.9174	0.8591	0.8917	0.7834	0.7012

proposed algorithm is little worse than FTQ and Informap algorithm. For the two real-life datasets without ground truth labels, SFI network and the Netscience network, the proposed algorithm outperforms the other algorithms, too.

In order to give a time cost analysis of the proposed algorithm, an experiment on the time comparison of three nature inspired algorithms is carried out in Figure 7. We can see that DPSO and the proposed algorithm have comparative time cost, but GA-net is the most time-consuming since the time complexity of decoding step is more than that of the others.

A further analysis on the effect of the proposed definition (impact of the node) and the adopted objective function is carried out on real-life networks. Figure 8 shows the experiment result obtained from different initialization and updating strategies; (a)~(d) represent the results of the proposed initialization and the proposed updating strategy, the proposed initialization and the updating strategy in [25], the initialization strategy in [25] and the proposed updating strategy, the initialization in [25] and the updating strategy in [25], respectively. When comparing (a) to (b) and (c) to (d), we can clearly see that the updating strategy has a better performance than the other strategy. The designed initialization strategy shows a weak superiority effect compared to other initialization strategies when comparing (a) to (c) and (b) to (d), which indicates that the proposed initialization strategy can generate a set of solutions with better quality than others. From this figure, we can also derive that the proposed algorithm have a faster convergence speed than the other strategies.

In order to analyse the benefit of the adopted modularity density objective function, we give a comparison experiment

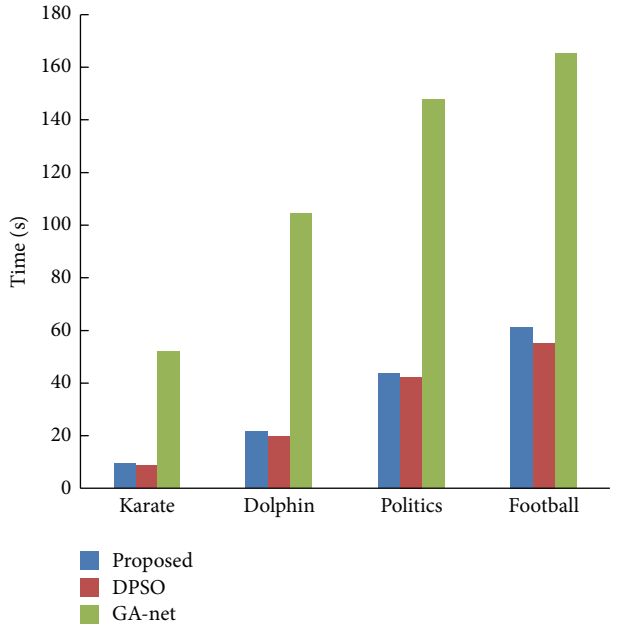


Figure 7: Computational time of three algorithms.

obtained from two different objective functions: modularity density and community score (GA-net adopted) in Figure 9. It shows that when we use the designed initialization and updating strategy, modularity density function is more effective than community score function.

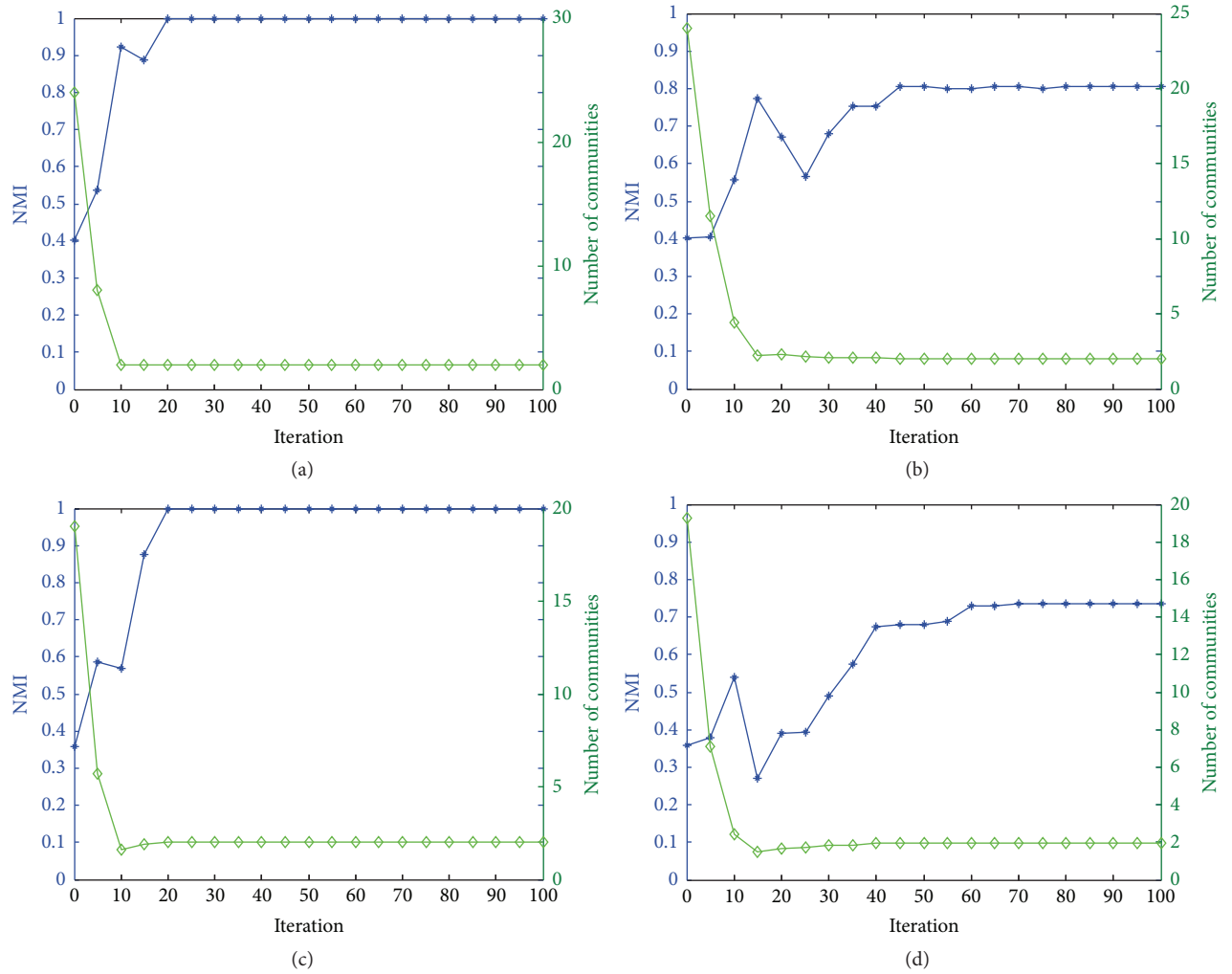


Figure 8: The comparison of different initialization and update strategies (the curve marked in green is the average number of communities and that in blue is the average NMI values during the optimization process).

Figure 10 displays a visible detection result obtained from the proposed algorithm on Zacharys Karate Club network. As is shown in this figure, the proposed algorithm figures out the true partition of the network when $\lambda = 0.3$. In the partition with $\lambda = 0.8$, our algorithm detects 4 communities, which is still reasonable since the original one community is divided into two subcommunities.

4.5. Experiments on Signed Networks. In this subsection, we apply the proposed algorithm to three real-world signed networks, including the illustrative signed network [34], the Slovene Parliamentary Party (SPP) network [35], and the Gahuku-Gama Subtribes (GGS) network [36].

Only DPSO and the proposed algorithm are considered on the three real-world signed networks because of the limitation of objective functions in the other algorithms. The best average statistical result is selected to be shown in Table 3.

As is shown in Table 3, we can clearly notice that both DPSO and the proposed algorithm can successfully detect the community structure of the network (NMI = 1) on SPP and GGS, but for the Illustrate network, the proposed algorithm shows a better performance than DPSO.

Table 3: Experiment results on signed networks.

Dataset	Proposed		DPSO	
	NMI	SQ	NMI	SQ
SPP	**1**	**0.4547**	1	0.4547
GGS	**1**	**0.4310**	1	0.4310
Illustrate	**1**	**0.5643**	0.8900	0.5406

SPP consists of 10 Slovene Parliamentary Parties set up by a series of experts on parliamentary activities in 1994. The topology community structure of the SPP network recognized by our algorithm is shown in Figure 11, in which the network is divided into two subcommunities when $\lambda = 0.3$.

GGS consists of 16 nodes which represent 16 Gahuku-Gama Subtribes involved in the warfare distributed in a particular area. The detection results of the proposed

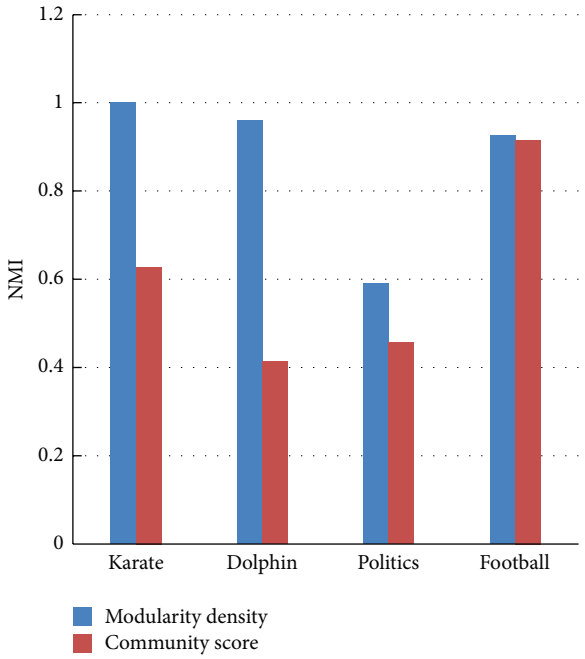
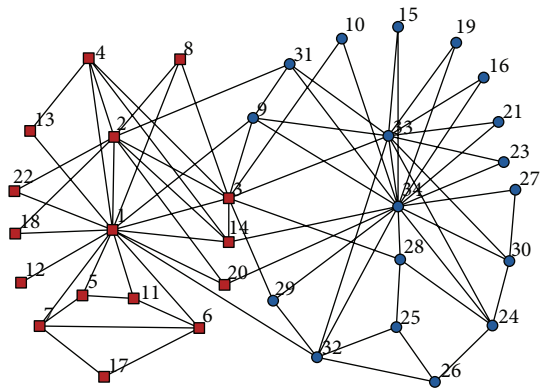
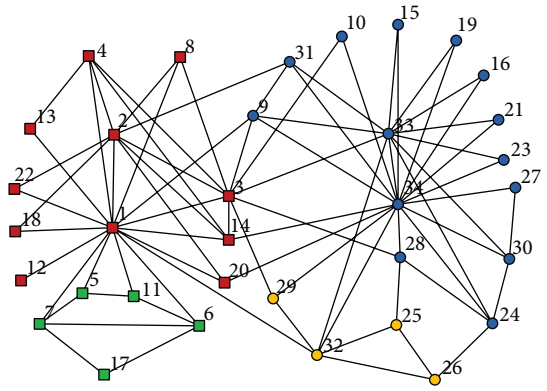


Figure 9: Benefit of modularity density.



(a) $\lambda = 0.3$



(b) $\lambda = 0.8$

Figure 10: Community structure detected by the proposed algorithm on Zacharys Karate Club.

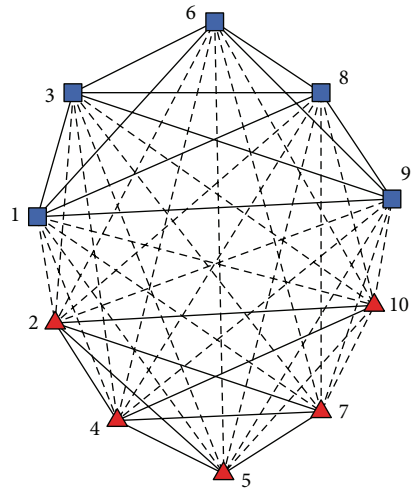


Figure 11: Community structure detected by proposed algorithm on SPP.

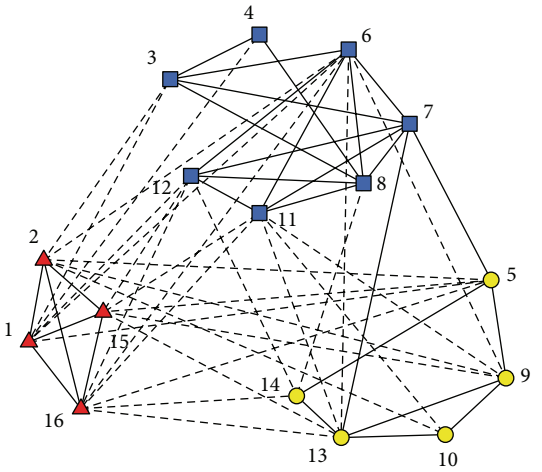


Figure 12: Community structure detected by the proposed algorithm on GGS.

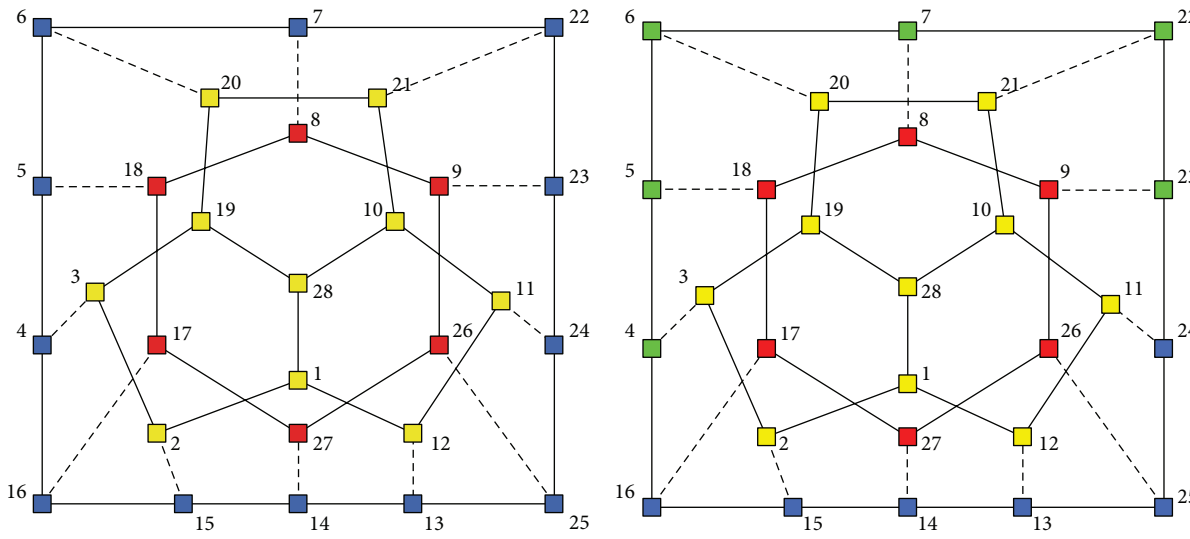
algorithm are displayed in Figure 12. Our algorithm detects three communities when $\lambda = 0.3$.

The illustrative signed network consists of 28 nodes and is divided into three communities. The community structure of it detected by the proposed algorithm is different when the value of parameter λ differs, in which the network is divided into three subcommunities with $\lambda = 0.2$ and four subcommunities with $\lambda = 0.3$. The corresponding result is shown in Figure 13.

From all the above analysis, the results on synthetic and real-life networks show that the proposed algorithm can deal with community detection problems effectively and promisingly.

5. Conclusion

After showing detailed description and experiments analysis in this paper, a conclusion about it can be drawn.



(a) $\lambda = 0.2$ (b) $\lambda = 0.3$

FIGURE 13: Community structure detected by the proposed algorithm on Illustrate.

Firstly, the definition of the impact of node is proposed and the new label propagation based on it is designed. This new definition demonstrates its effectiveness by experiments on synthetic and real-life networks.

Secondly, special initialization and updating strategies based on the impact of node are designed. By using these strategies, the structure of the community is detected more exactly compared with other methods.

Thirdly, we modify modularity density function to signed networks according to the character of networks. Combining proposed initialization and updating strategies with this objective function, it can detect the community structure of signed networks exactly.

Additionally, the proposed strategy can be implemented into other graph-related fields easily. This paper adopts single-objective function, modularity density, to detect community structure, which can be easily extended to a multiobjective optimization problem. We believe that it would show its superiority to others in this way. Moreover, applying community detection to our radar communication networks to improve the cooperation quality is also one of our future works.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] J. Hopcroft, O. Khan, B. Kulis, and B. Selman, "Natural communities in large linked networks," in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 541–546, ACM, Washington, DC, USA, August 2003.

[2] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '99)*, pp. 251–262, ACM, Cambridge, Mass, USA, August-September 1999.

[3] S. Lozano, J. Duch, and A. Arenas, "Analysis of large social datasets by community detection," *The European Physical Journal Special Topics*, vol. 143, no. 1, pp. 257–259, 2007.

[4] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.

[5] C. A. Hidalgo, B. Winger, A.-L. Barabási, and R. Hausmann, "The product space conditions the development of nations," *Science*, vol. 317, no. 5837, pp. 482–487, 2007.

[6] M. Chen, K. Kuzmin, and B. K. Szymanski, "Community detection via maximization of modularity and its variants," *IEEE Transactions on Computational Social Systems*, vol. 1, no. 1, pp. 46–65, 2014.

[7] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.

[8] U. Elsner, "Graph partitioning—a survey," in *Encyclopedia of Parallel Computing*, pp. 805–808, Springer US, 1997.

[9] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 6, Article ID 066133, 5 pages, 2004.

[10] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, no. 6, Article ID 066111, 2004.

[11] R. Guimerà and L. A. N. Amaral, "Functional cartography of complex metabolic networks," *Nature*, vol. 433, no. 7028, pp. 895–900, 2005.

[12] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 72, no. 2, Article ID 027104, 2005.

[13] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 23, pp. 8577–8582, 2006.

[14] C. Pizzuti, "GA-Net: a genetic algorithm for community detection in social networks," in *Parallel Problem Solving from Nature—PPSN X*, pp. 1081–1090, Springer, Berlin, Germany, 2008.

[15] C. Pizzuti, "A multiobjective genetic algorithm to find communities in complex networks," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 3, pp. 418–430, 2012.

[16] A. Arenas and A. Díaz-Guilera, "Synchronization and modularity in complex networks," *The European Physical Journal: Special Topics*, vol. 143, no. 1, pp. 19–25, 2007.

[17] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 1, pp. 36–41, 2007.

[18] Z. Li, S. Zhang, R.-S. Wang, X.-S. Zhang, and L. Chen, "Quantitative function for community detection," *Physical Review E*, vol. 77, no. 3, Article ID 036109, 2008.

[19] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science (MHS '95)*, vol. 1, pp. 39–43, Nagoya, Japan, October 1995.

[20] C. A. Coello Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.

[21] L. C. Cagnina, S. C. Esquivel, and C. A. C. Coello, "A fast particle swarm algorithm for solving smooth and non-smooth economic dispatch problems," *Engineering Optimization*, vol. 43, no. 5, pp. 485–505, 2011.

[22] Z. Zhu, J. Zhou, Z. Ji, and Y.-H. Shi, "DNA sequence compression using adaptive particle swarm optimization-based memetic algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 643–658, 2011.

[23] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation*, vol. 5, pp. 4104–4108, Orlando, Fla, USA, October 1997.

[24] W.-N. Chen, J. Zhang, H. S. H. Chung, W.-L. Zhong, W.-G. Wu, and Y.-H. Shi, "A novel set-based particle swarm optimization method for discrete optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 2, pp. 278–300, 2010.

[25] M. Gong, Q. Cai, X. Chen, and L. Ma, "Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 82–97, 2013.

[26] D. Aloise, G. Caporossi, P. Hansen, L. Liberti, S. Perron, and M. Ruiz, "Modularity maximization in networks by variable neighborhood search," in *Proceedings of the 10th DIMACS Implementation Challenge Graph Partitioning and Graph Clustering*, pp. 113–127, Atlanta, Ga, USA, February 2012.

[27] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 4, pp. 1118–1123, 2007.

[28] S. Gómez, P. Jensen, and A. Arenas, "Analysis of community structure in networks of correlated data," *Physical Review E*, vol. 80, no. 1, Article ID 016114, 2009.

[29] F. Wu and B. A. Huberman, "Finding communities in linear time: a physics approach," *The European Physical Journal B—Condensed Matter and Complex Systems*, vol. 38, no. 2, pp. 331–338, 2004.

[30] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, no. 4, Article ID 046110, 2008.

[31] W. Zachary, "An information flow modelfor conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977.

[32] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.

[33] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, Article ID 036104, 2006.

[34] B. Yang, W. K. Cheung, and J. Liu, "Community mining from signed social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 10, pp. 1333–1348, 2007.

[35] A. Ferligoj and A. Kramberger, *An Analysis of the Slovene Parliamentary Parties Network*, 1996.

[36] K. E. Read, "Cultures of the central highlands, new Guinea," *Southwestern Journal of Anthropology*, vol. 10, no. 1, pp. 1–43, 1954.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 5725143, 15 pages
http://dx.doi.org/10.1155/2016/5725143

Research Article

An Efficient Stock Recommendation Model Based on Big Order Net Inflow

Yang Yujun,[1,2,3] Li Jianping,[1] and Yang Yimei[2]

[1]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
[2]*Department of Computer Science and Technology, Huaihua University, Huaihua 418008, China*
[3]*Hunan Provincial Key Laboratory of Ecological Agriculture Intelligent Control Technology, Huaihua 418008, China*

Correspondence should be addressed to Yang Yimei; yym@hhtc.edu.cn

Received 14 August 2015; Revised 10 December 2015; Accepted 15 December 2015

Academic Editor: David Bigaud

In general, the stock trend is mainly driven by the big order transactions. Believing that the stock rise with a large volume is closely associated with the big order net inflow, we propose an efficient stock recommendation model based on big order net inflow in the paper. In order to compute the big order net inflow of stock, we use the M/G/1 queue system to measure all tick-by-tick transaction data. Based on an indicator of the big order net inflow of stock, we select some stocks with the higher value of the net inflow to constitute the prerecommended stock set for the target investor user. In order to recommend some stocks with which this style is familiar them to the target users, we divide lots of investors into several categories using fuzzy clustering method and we should do our best to choose stocks from the stock set once operated by those investors who are in the same category with the target user. The experiment results show that the recommended stocks have better gains during the several days after the recommended stock day and the proposed model can provide reliable investment guidance for the target investors and let them get more stock returns.

1. Introduction

In the area of stock recommendation method research [1], most of the research mainly focuses on the two areas: stock recommendation methods based on stock comment [2] and price forecasting [3]. The former is easy to understand and master for investors. However, in such a complicated stock market, the investors do not know which one to believe among the lots of stock comments with dubious authenticity and every choice has a great risk for them. The latter method [4] is difficulty for investors to understand and master since the application of the latter method is relatively complex and involves a lot of profound mathematical knowledge [5]. Given this situation, many scholars have done a lot of research on the stock recommendation [6].

Currently, the stock recommendation based on price forecasting relies [7] mainly on mathematical and statistical methods [8], time series model [9, 10], and machine learning model [11]. Sonsino and Shavit [12] have researched a stock prediction and selection method based on unidentified historical data. M.-Y. Chen and B.-T. Chen [13] have proposed a stock price forecasting method based on the hybrid fuzzy time series and granular computing. Xin et al. [14] have given a strategy for filtering out users with similar demand characteristics by using collaborative recommendation algorithm with fuzzy clustering method, which shows excellent recommendation effect.

In present financial field, how to integrate multiple technologies [15], such as data mining, machine learning and herd psychology [11], and other nontraditional technologies, into stock recommendation has become a hot topic. Few papers use money net inflow as stocks recommendation techniques. Given this situation, we proposed an efficient stock recommendation model based on big order net inflow in the paper. At first, we divide lots of users into several categories utilizing collaborative filtering algorithm based on user fuzzy clustering [16]. We get some stocks from the stocks once operated by those users in same category and form a

prerecommended stock set. Then, we use a method based on M/G/1 [17] to compute the net inflow amount of big order for every stock in the prerecommended stock set. From the prerecommended stock set descending ordered by the value of big order net inflow, we choose some stocks with highest value in front of the set as the last recommendation stock set for the target user.

In general, we believe that the stock rise with a large trading volume is closely related to the purchase stock of big order. In order to analyze the big order net flow of stock, we need to observe the stock trading volume and turnover. The money net flow and the money flow [18] are different in concept. The big order refers to the amount of each transaction over one million yuan or the volume of each transaction over fifty thousand in a single transaction. So the big order net inflow refers to the amount of money of big order buy or sell of the same stocks within a day. Most of the time, the money flow is bigger than zero, and the money net flow is less than zero. In individual cases, the money net flow is bigger than zero, and the big order net flow is less than zero. Under this situation, we proposed an efficient stock recommendation model based on big order net inflow. The new recommendation model can measure the capital and the pulsation of the stock markets and consider investors preferences and behavior characteristics; it can improve the existing deficiencies of some current stock recommendation. In addition, the new recommendation model can analyze and filter the stock with less returns in the future and improve the investment gains of investors. The experiment results show that the recommended stocks have better gains during the several days after the recommended stock day and the proposed model can provide reliable investment guidance for the target investors and let them get more investment returns.

The rest of this paper is organized as follows. Section 2 briefly reviews the definitions and theorems of fuzzy clustering and the framework of the collaborative filtering algorithm based on user fuzzy clustering. Section 3 demonstrates the method for computing big order net inflow and the framework of the proposed model. Section 4 presents the simulation experiment and empirical analyses of the proposed model; finally some conclusions are given and some future works are pointed out in Section 5.

2. Theoretical and Modeling Framework

The concept of fuzzy set [16] in fuzzy cluster is put forward by Zadeh in 1965. Gath and Bar-On earlier applied that theory to compute the scoring of poly graphic sleep recordings in their study [19]. In this section, we will briefly review the definitions and theoretic of fuzzy cluster.

2.1. Fuzzy Clustering Theory

Definition 1. Defined set $R = (r_{ij})_{m\times n}$, a matrix with m rows and n columns; if $0 \le r_{ij} \le 1$, then R is called fuzzy matrix R. When r_{ij} is only 0 or 1, said R is a Boolean matrix. When elements r_{ii} of the diagonal are all 1 in fuzzy matrix $R = (r_{ij})_{n\times n}$, said R is reflexive fuzzy matrix.

Definition 2. If A is n-order square, defined $A^2 = A \circ A$, $A^3 = A^2 \circ A, \ldots$, and $A^k = A^{k-1} \circ A$.

Definition 3. Defined set $R = (r_{ij})_{m\times n}$, and called $R^T = (r_{ij}^T)_{m\times n}$ is the transposed matrix of R, where $r_{ij}^T = r_{ij}$.

Definition 4. Defined set $R = (r_{ij})_{m\times n}$, for any $\lambda \in [0, 1]$, and said $A_\lambda = (r_{ij}^{(\lambda)})_{m\times n}$ is the λ-cut matrix of the fuzzy matrix R. When $r_{ij} \ge \lambda$, $r_{ij}^{(\lambda)} = 1$ and $r_{ij} < \lambda$, $r_{ij}^{(\lambda)} = 0$, said A_λ is a Boolean matrix, and said λ is a confidence level parameter or cut level parameter.

Definition 5. Suppose two limited discourse domains $X = \{x_1, x_2, \ldots, x_m\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$; then X to Y fuzzy relation R is an $m \times n$ order fuzzy matrix, and set $R = (r_{ij})_{m\times n}$, where $r_{ij} = R(x_i, y_j) \in [0, 1]$ represents (x_i, y_j) relevance on fuzzy relation R.

Theorem 6. *If R is fuzzy similar matrix, then, for any natural number k, R^k is fuzzy similarity matrix.*

Theorem 7. *If R is an n order fuzzy similar matrix, then there is minimum natural number k ($k \le n$), for all natural number m ($m \ge k$); constant $R^m = R^k$, R^k, namely, fuzzy equivalent matrix ($R^{2k} = R^k$). At this point, said R^k is the transitive closure of R, denoted by $t(R) = R^k$.*

Method 8. If R is fuzzy similar matrix, then the following method will solve the transitive closure $t(R) : R \to R^2 \to R^4 \to R^8 \to R^{16} \to \cdots \to R^{2^k}$, and the said method is the square self-synthesis method (S^3M).

2.2. Fuzzy Clustering Analysis. The fuzzy clustering analysis is an analyzing clustering and classification method by establishing fuzzy similar relationship of objective things based on the objective characteristics, the degree of closeness, and similarity between objective things.

In Figure 1, the fuzzy clustering processing model can be divided into four stages, namely, the data preprocessing, the data standardization, the constructing fuzzy similar matrix (FSM), and the clustering and analysis.

2.2.1. Data Preprocessing. In China, it is well known that individual investors buy or sell stocks only through a securities company, not directly from the stock exchange. In order to trade stocks, individual investors firstly have to register as a member in a securities company. According to the National Security Act, investors must have a test in the risk tolerance when investors are registering. The securities company gives them a risk test paper with fifteen questions or more. Each question has four or five options for investors to choose. In order to store and process, we use an integer value instead of the chosen answer of investors in the user risk database. In this paper, we choose twelve answers among all answers to study. There are at least 1 million records about risk information of investors in the user risk database of any securities company because there

Table 1: The surveying contents of investor.

x_{ij}	Denoted contents	Range
x_{i1}	The age range of investor	[1, 5]
x_{i2}	The funds amount of investor that can be used to invest	[1, 5]
x_{i3}	The investment purposes of investor	[1, 4]
x_{i4}	The duration of your general investment	[1, 4]
x_{i5}	The maximum loss that you can accept within one year	[1, 5]
x_{i6}	The selection of four portfolio average returns with the best and worst of earnings in the next three years	[1, 4]
x_{i7}	The annual income change of your family in next five years	[1, 4]
x_{i8}	The ratio of domestic consumption expenditure share of total income monthly in your family	[1, 5]
x_{i9}	The highest risk products that you have invested before	[1, 4]
x_{i10}	The annual income of your family	[1, 5]
x_{i11}	The number of years since you invest	[1, 4]
x_{i12}	The type of your work	[1, 5]

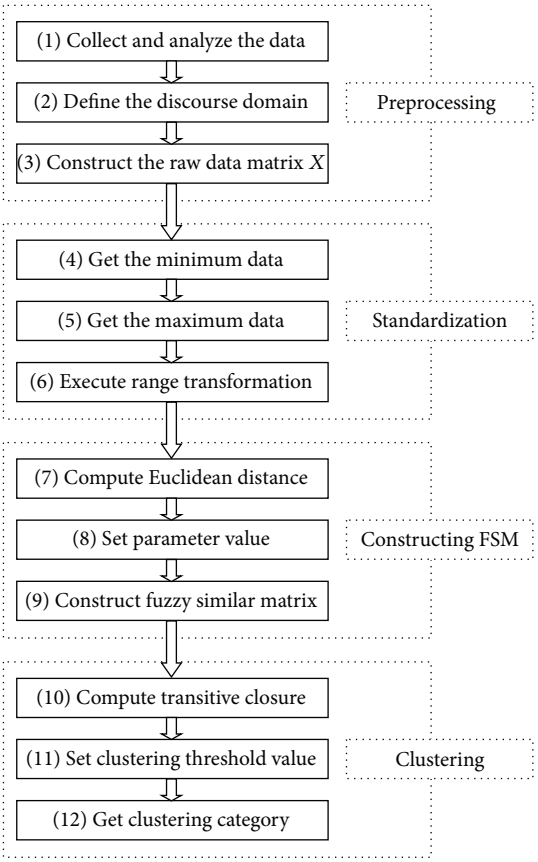


Figure 1: The structure of fuzzy clustering processing model.

are more than 200 million stocks investors in China. So we assume that an investor risk tolerance surveying database exists, which includes n investors risk tolerance surveying data. The characteristics of these data which include the following twelve contents reflect the investment style and risk tolerance of the investors. We define the set of investor risk tolerance surveying data as $X = \{x_1, x_2, \ldots, x_n\}$, where x_i denotes the surveying data of the ith investor in the set X, which is constituted by the following contents vector in a fixed order $x_i = \{x_{i1}, x_{i2}, x_{i3}, \ldots, x_{ij}, \ldots, x_{i12}\}$, where j belongs to [1, 12]. Table 1 shows the detailed contents of the ith investor in the investor risk tolerance surveying database.

2.2.2. Data Standardization. We firstly compute all columns data of raw matrix X, get the minimum data and the maximum data of each column, and then compress each data of matrix X to [0, 1] using the following transformation formula. After the above standardizing process, we can get standardized matrix Y with standardizing data:

$$
\begin{aligned}
&Y = \left\{y_{ij}\right\}, \quad 1 \le i \le n,\ 1 \le j \le 12,\\
&y_{ij} = \frac{x_{ij} - \min_{1\le i\le n}\left\{x_{ij}\right\}}{\max_{1\le i\le n}\left\{x_{ij}\right\} - \min_{1\le i\le n}\left\{x_{ij}\right\}}, \quad 1 \le j \le 12,\\
&Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1,12} \\ y_{21} & y_{22} & \cdots & y_{2,12} \\ \vdots & \vdots & & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{n,12} \end{bmatrix}.
\end{aligned}
\tag{1}
$$

2.2.3. Constructing Fuzzy Similar Matrix. After obtaining standardized matrix Y with standardizing data, we use the direct Euclidean distance method as the similarity coefficient method to determine the similarity coefficient among investors and construct the fuzzy similar matrix. Consider

$$
r_{ij} = 1 - c \times d\left(x_i, x_j\right), \tag{2}
$$

where c is a suitable choice of parameters so as to $0 \le r_{ij} \le 1$ and the $d(x_i, x_j)$ represents the distance between x_i and x_j as follows:

$$
\begin{aligned}
&d\left(x_i, x_j\right) = \sqrt{\sum_{k=1}^{12}\left(x_{ik} - x_{jk}\right)^2},\\
&c = \frac{1}{1 + c'},
\end{aligned}
\tag{3}
$$

where c' belongs to $[0, \infty)$ and c belongs to $(0, 1]$. We can choose any value for c'. If the c' value is too high, parameter c will be less, which will lead to increasing the accuracy of computing. Therefore, we chose an appropriate value for c'. In fact, we can choose zero value for c' which does not affect the experiment results; c is equal to 1.

Thus, we can get fuzzy similar matrix R of n by n between investors:

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{bmatrix}. \tag{4}$$

2.2.4. Clustering and Analysis. Since fuzzy similar matrix R is a n-dimensional square matrix, we can carry out transitive closure $t(R)$ using the square self-synthesis method:

$$t(R) : R \longrightarrow R^2 \longrightarrow R^4 \longrightarrow R^8 \longrightarrow R^{16} \longrightarrow \cdots \longrightarrow R^{2^k}, \tag{5}$$

where $k \leq [\log_2^n]$.

According to the actual situation, we have to choose an appropriate λ value between 0 and 1 for λ-cut matrix $t(R)_\lambda$ of transitive closure $t(R)$ of fuzzy matrix R. Having a classification of $t(R)$ based on λ, we can get equivalence classification matrix $t(R)_\lambda$ under the given λ value.

Thus, we will get clustering category for investors. One day, a new investor who becomes the target recommendation investor of stock recommendation system will be added to the investor risk tolerance surveying database. In order to determine which category the new investor belongs to, we utilize the fuzzy clustering method to subdivide the database into several groups based on the above λ-cut value.

2.3. Collaborative Filtering Algorithm

2.3.1. Nearest Neighbors Choice. Currently, the collaborative filtering algorithm is the most successful personalized recommendation algorithm. It can be classified into two categories: one is item-based collaborative filtering algorithm [20]; the other is user-based collaborative filtering algorithm. The former was first put forward by Sarwar et al. [21] that when we calculate the user similarity, first we calculate the similarity between items to select the most similar items and then predict the rating. The latter was first proposed by Goldberg et al. [22], which is according to the rating of target user's nearest neighbors to predict the target item rating. This algorithm can be computed offline, shorten the time of online calculation, and increase speed of the online recommendation.

In order to improve the accuracy of the nearest neighbor investor's choice, we will use the user-based collaborative filtering algorithm to compute the degree of similarity in behavior between other investors and the target investor in the same fuzzy cluster. Then, according to the stocks of the nearest investors and target investor, we can generate optimized stock list from the stock set calculated by fuzzy cluster algorithm and choose the top-k stock to recommend the target investor.

At first, we need to calculate the degree of similarity between investors based on their risk tolerance surveying data. At present, there are several methods to calculate the similarity between investors, such as cosine-based similarity, the adjusted-cosine similarity, and Person correlation-based similarity [23]. According to the degree of similarity between the target investor and other investors, we can generate neighbor set $U = \{U_1, U_2, U_3, \ldots, U_n\}$ for the target investor u. Then, we choose the top-k investors as neighbor investor for the target investor based value of $\text{sim}(u, U_i)$ ordered by descending.

2.3.2. Constructing Stock Cluster Set. In order to recommend some more accurate stocks to the target investor, we subdivide the stocks into several categories as follows based on the fuzzy clustering method in Section 2.2 in this paper. Then, we can utilize fuzzy clustering analysis method to construct stock cluster set. As we know, there are dozens of attributes in each stock. We choose six important attributes of stock to construct a stock attributes database. We think the six attributes of stock are the most important attributes for stock clustering. The six attributes are the daily average gains, the daily average amplitude, the days of price rise, the net profit in last year, the daily net amount of big order, and the days with net amount of big buying order. According to the six attributes of stock and the data format like Table 1 in this paper, we construct the stock attributes database. Then, we cluster the stocks in the stock attributes database and subdivide the stocks into several categories. We can effectively distinguish different stocks between poor stocks and good stocks. Such clustering results can reflect those stocks operational characteristics and be able to provide more accurate and effective recommendation information for target investor. In generally, the stocks in same cluster have similar trading dynamic characteristic. If we have a recommended stock for target investors, we will try to find some other stocks in its stock cluster set. This can improve the accuracy of recommended stocks and reduce the difficulty of the search for other recommended stocks. In order to reduce the computational complexity of clustering the stocks, we have to guarantee that the number of cluster is less than 10. The number of cluster will vary with the changes of recommended stocks. Generally, the number of cluster varied between five and seven in experimentation.

2.3.3. Generating Recommended Stocks List. At first, we get the stocks list of the target investor, and then we get the score of each stock in stocks list of target investor and the score of same stock like in stocks list of neighbor investors. If some stocks are not in the stocks list of neighbor investors, we can forecast their scores using the following formula:

$$f(u, j) = \frac{\sqrt{\sum_{k=1}^{K} \text{sim}(i, k)^2 * \left(S_k - \overline{S_k}\right)^2}}{\sqrt{\sum_{k=1}^{K} \text{sim}(i, k)^2}} + \overline{S_u}, \tag{6}$$

where $\overline{S_u}$ represents the target investor u's average scores for all stocks, S_k represents the kth neighbor investor scores of the target investor u, $\overline{S_k}$ represents the average scores of k neighbors investor for all stocks, and $f(u, j)$ represents the jth stocks forecast score of target investor u. $\text{sim}(i, k)$ is a modified similarity formula based on the cosine Similarity; it is defined as follows:

$$\text{sim}(i,k) = \frac{\sum_{v=1}^{m} \left(R_{vi} - \overline{R_i}\right)\left(R_{vk} - \overline{R_k}\right)}{\sqrt{\sum_{v=1}^{m} \left(R_{vi} - \overline{R_i}\right)^2} \sqrt{\sum_{v=1}^{m} \left(R_{vk} - \overline{R_k}\right)^2}}, \quad (7)$$

where $\overline{R_i}$ is the ith stock average scores for all investors, $\overline{R_k}$ is the kth stock average scores for all investors, R_{vk} represents the kth neighbor investor scores for the vth stock, and R_{vi} represents the ith neighbor investor scores for the vth stock.

Now we give a simple example for the process of generating recommendation stock list. Assuming that there are two stocks in the stocks list of the target investor, such as stock A and stock B, here we write them as a set $\{A, B\}$. Then we get the stocks set of four or more neighbors, such as set $\{B, C, D\}$, $\{B, C, E\}$, $\{C, F\}$, and $\{B, C\}$, and we can select the rated top-k stocks as the target investor's extend stocks list from the above five stock set. When we set k as 2, we can get the top-2 stocks set $\{B, C\}$.

Secondly, in order to get the more large stock extend set of the target investor, such as stock set $\{B, C, X, Y, Z\}$, we need to revise the k value. If we recommend to the target investor only five stocks, we will stop revising the k value when the number of stock in the above stock set is greater than or equal to five.

Finally, we calculate net inflow amount of big order for each stock of the above stock extend set $\{B, C, X, Y, Z\}$ recently, then we sort the five stocks in descending order according to the net inflow amount of big order and select top-n stocks recommend to the target investor. If we set n as three, we get the recommend stock list $\{B, X, Y\}$. Because the process of judging money inflow is more complex, we propose a new method based on M/G/1 to compute the net inflow amount of big order and we can forecast the direction of stock price movements in the future.

2.4. M/G/1 Queue System. The M/G/1 queue system [24] has been extensively studied for the last three decades [25]. According to the circumstances of the securities transaction, here we review the single server queue system which behaves like the usual M/G/1 queue when the server is working. We assume that the server goes through cycles of idle and busy periods. The idle periods include two cases: one is when there is no work to do, and the other is when there is work to do but the server is on vacation. The busy periods are the times when the server is actually working on the customers of primary customers [26]. During the busy periods of the simple M/G/1 queue system, we assume that the customers arrival time follows Poisson distribution with parameter λ. Let λ be the customer arrival rate and let D be the distribution of the service times of customers arriving during busy periods; then the customers arrival time has a general distribution function B. Let S_i be the epoch at the end of the ith busy period and let T_i be the epoch beginning the ith busy period. Then, $0 = T_i < S_i < T_{i+1} < S_{i+1} < \cdots < T_n < S_n$. We assume that the arrival process and the service times of the customers arriving in the interval (T_i, S_i) are independent of those arriving in (T_j, S_j) for $i \neq j$. Let $w(t)$ be the work in the queue system at time t. Let $W_{1i} = w(S_i)$ and $W_{2i} = w(T_{i+1})$, where $i > 0$. Then, W_{1i} is the work in the queue system at the end of the ith busy period and W_{2i} is the work in the queue system immediately after the beginning of the $(i + 1)$st busy period.

Let $l(k)$ be the kth workload step in all workload process. Let $L(k) = \min\{m : \sum_{i=1}^{m}(S_i - T_i) \geq k\}$ and $E(k) = \sum_{i=1}^{L(k)-1}(T_{i+1} - S_i)$.

If $k < \sum_{i=1}^{L(k)}(S_i - T_i)$, then $l(k) = w(k + E(k))$.

If $k = \sum_{i=1}^{L(k)}(S_i - T_i)$, then $l(k^+) = w(k + E(k) + T_{L(k)+1} < S_{L(k)})$ and $l(k^-) = w(k + E(k))$.

Clearly, if $k = \sum_{i=1}^{L(k)}(S_i - T_i)$, then $l(k^+) - l(k^-) = W_{2,L(k)} - W_{1,L(k)}$. The $l(k)$ process behaves like the work in a simple M/G/1 queue.

3. Proposed Method

This section consists of two subsections: Section 3.1 describes the method based on M/G/1 to compute the net inflow amount of big order and Section 3.2 describes the proposed model and method for stock recommendation.

3.1. Compute the Net Inflow Method

3.1.1. Funds Flow Theory. The flow of funds is stock movement direction actively chosen by the funds in the stock market. From the amount of perspective to analyze the flow of funds, namely, observation volume and turnover, trading volume and turnover in the actual operation is directional, to buy or sell. For both the stock market trend analysis and the operation on individual stocks, the determination of the funds flow plays a vital role, and the process of the funds flow is more complicated, not easy to grasp. The funds flow can help investors see what others are doing in the end through the index (price) change fog. For example, the index (price) of a stock rise up to a point may be driven by 10 million funds or a billion of funds, both of which have a completely different significance for investors. In general, funds flow and the trend of stock index change are very similar, but in the following two cases funds flow measure has obvious significance. One is that the day's flow of funds and stock index change opposite. For example, the stock overall index is down throughout the day, but funds flow shows a positive net inflow of funds throughout the day. The other is that there is very big opposite between the funds flow and the stock index change. For example, the stock index rises highly throughout the day, but the actual net inflows are small or even negative which is called net outflows. When the funds flows and the stock index change is opposite, the funds flow can reflect stock actual movement direction more than the stock index change in the future.

3.1.2. Funds Flow Concepts. In order to more clearly describe the proposed method, we review some concept about funds flow as follows.

Definition 9. Funds inflow refers to the amount of active buying. It is active buying transactions where the buyer actively buys stock with price equal to or higher than the first selling price.

Definition 10. Funds outflow refers to the amount of active selling. It is active selling transactions where the seller actively sells stock with price equal to or less than the first buying price.

Definition 11. Funds net inflow refers to the amount of funds inflow minus funds outflow. If it is positive, the probability of stock price rise is higher than fall and vice versa if the net inflow is negative; then, the probability of stock price fall is higher than rise.

Definition 12. Big order refers to the buying or selling order with big amount. According to the amount of the order, we divide it into four categories: small order, general order, big order, and very big order (king order). Using the number of the order which measures it, we think that the number of big order is more than 5 million shares or more than 20 million yuan and the number of king order is more than 20 million shares or more than 100 million yuan.

Definition 13. Big order net amount refers to the difference between the amount of big buying order and the amount of big selling order. If it is positive, we call it the big order net inflow. If not, we call it the big order net outflow.

Definition 14. Tick-by-tick data refer to the single transaction during transactions. It reflects the true circumstances of the transaction process and is proprietary data of the Level-2.

3.1.3. Method Based on M/G/1. We review the classical M/G/1 queue system which has the following four characteristics [27].

(1) The characteristic of the arrival process follows Poisson distribution with arrival rate parameter λ; M indicates a Poisson process without memory.

(2) Probability P of the service time follows general random distribution. Let t_i be the service time for the ith customer that is an independent and random value and has general distribution function $B(t)$:

$$P\left(t_i \geq t\right) = B\left(t\right), \quad i = 1, 2, 3, \ldots. \tag{8}$$

The mean value and variance of service time are given by

$$\begin{aligned} \overline{X} &= \frac{1}{u} = \int_0^\infty t dB\left(t\right), \\ \overline{X^2} &= \int_0^\infty t^2 dB\left(t\right). \end{aligned} \tag{9}$$

(3) The number of server desk is one; the arrival time and service time are independent of each other.

(4) The system allows an infinite captain for the length of customers; the queue discipline is first come first serve (FCFS).

Assuming that interarrival time follows Poisson distribution with parameter λ in $\{A(t),\ t \geq 0\}$, let $A(t)$ be the number of arrival customers in $[0, t]$ time, let $N(t)$ be the number of customers in queue system at time t, let X_n be the number of customers after the nth customer departure instant, let d_n be the departure time of the nth customer, and let a_n be the arrival time of the nth customer. If the number of customers is greater than zero at time d_n, then

$$X_{n+1} = X_n + A\left(d_{n+1}\right) - A\left(d_n\right) - 1 \tag{10}$$

or

$$d_{n+1} = d_n + t_n + 1. \tag{11}$$

At this time if $X_n = 0$, then

$$X_{n+1} = A\left(a_{n+1} + t_{n+1}\right) - A\left(a_{n+1}\right). \tag{12}$$

From (11) and (12) the following can be obtained:

$$P\left\{X_{n+1} = i \mid X_1, X_2, \ldots, X_n\right\} = P\left\{X_{n+1} = i \mid X_n\right\}. \tag{13}$$

Transition probability matrix P of Markov Chain X_n is

$$P = \begin{pmatrix} p_0 & p_1 & p_2 & p_3 & \cdots \\ p_0 & p_1 & p_2 & p_3 & \cdots \\ 0 & p_0 & p_1 & p_2 & \cdots \\ 0 & 0 & p_0 & p_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots \end{pmatrix}, \tag{14}$$

where chain X_n has Markov property and p_k is given by

$$p_k = \int_0^\infty \frac{e^{-\lambda t}\left(\lambda t\right)^k}{k!} dB\left(t\right), \quad \left(k = 0, 1, 2, \ldots\right). \tag{15}$$

3.2. The Framework of the Proposed Model. In Figure 2, the process and data flow framework of the proposed model can be divided into two stages, namely, the user clustering stage and stock recommend stage. Generally, investors who have similar characteristics have similar investment interest. According to Figure 2, in order to obtain accurately the similar stocks for target users, we have to process user clustering and obtain some similar users. The selection of the clustering threshold value affects the number and size of the user category and then affects the accuracy of the stocks set which is selected from the same category user stocks, so it is critical to select the value of user clustering threshold. In the stock recommend stage, the computing of the big orders net for lots of stocks is the most important part. The stock trend is mainly driven by big order transactions.

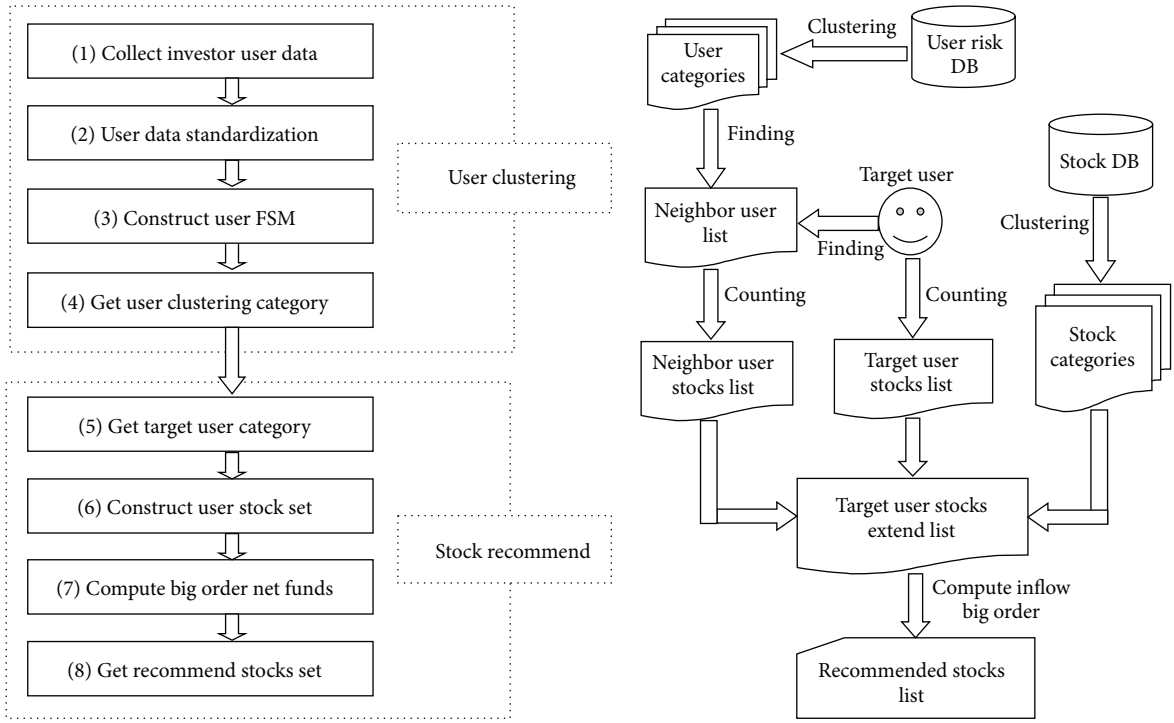


Figure 2: (a) The process framework of the proposed model. (b) The data flow framework of the proposed model.

It is generally believed that stocks rise with a large volume is closely associated with big orders net amount to buy, so the stocks are generally rising in price under the trend driven by big orders net inflow which is called big orders net buying. In contrast, the stocks are generally falling in price under the trend driven by big orders net outflow which is called big orders net selling. The big orders net amount includes the big orders net buying and the big orders net selling. The traditional model uses the funds inflow and funds outflow to predict stock trend that is unsuitable for some new special situation. For example, one day the funds inflow of a stock is far greater than zero, but the big orders net inflow of the stock is far smaller than zero. If it is that case, we can predict that this stock will go to falling trend over a period of time after that day. After observing many stocks, we found that it is indeed the case. Our proposed model can solve this problem by using big orders net amount that can avoid or reduce some forecast errors and can improve accuracy of recommend stocks trend in the future. Then, according to the indicated results of big orders net inflow, we select some optimal stocks recommend to target users to buy. Correspondingly, according to the indicated results of big orders net outflow, we select some optimal stocks recommend to target users to sell.

4. Simulation Experiment

In this section, we study and compare the performance of the proposed model. In general, during the Shanghai and Shenzhen Composite Index (CSI) rise, the accuracy of the recommendation algorithm is higher, but during the CSI fall, the accuracy of recommendation algorithm is very low or even completely incorrect. Thus, in the course of falling, the experimental results can test a recommendation algorithm. In order to make the experiment results more objective and realistic, we use stock return to test whether the recommended stock has a good return from 10 to 30 days after that. For target investor, the higher the yields the investor gets, the better the effect of the recommendation model.

Figure 3: The research interval price trends graph of the Shanghai Composite Index (CSI).

4.1. Data Selection. In order to examine whether the proposed model has made improvement in prediction accuracy, we select data at four different periods of the real stock market in China as the experiment data. The four periods include bottom period (2012/11/28–2012/12/04, see A in Figure 3), middle period (2012/12/17–2012/12/21, see B in Figure 3), top period (2013/02/04–2013/02/08, see C in Figure 3) of

Table 2: The users clustering results.

Category number	Users numbers	Percentage (%)
1	912	35.63
2	138	5.39
3	576	22.50
4	482	18.83
5	314	12.27
6	48	1.88
7	90	3.52

Table 3: The nearest neighbors of the target user.

No.	Similarity value	Users number
1	0.982	380
2	0.957	359
3	0.926	2478
4	0.903	1911
5	0.871	1653
6	0.819	406
7	0.735	141

a wave with rising trend, and middle period (2013/06/17–2013/06/21, see D in Figure 3) of a wave with falling trend. The experimental data include data of three parts that are the investor user data, the CSI data, and the tick-by-tick transaction data of stock. As the user's data are related to the user's personal privacy, we use the data which are removed from the user's privacy as the experimental data. The CSI and stocks data are stock market free data, but the tick-by-tick transaction data of stock are the Level-2 data and charging data.

4.2. Data Processing. In order to reduce the amount of computation and be suitable for parallel computing, we randomly select 2560 users from the investor user database to fuzzy clustering and divide those users into several categories according to the threshold of clustering. By the nature of the clustering, the value of the threshold can control the number of the user categories. The value of the threshold and the number of the user categories are inversely related. The threshold can be in the range from 0 to 1. If the threshold is set too high, we will get very few user categories. Contrarily, if the threshold is set too small, we will get much more user categories. For example, if the threshold is set to 1, we will get 2560 user categories. Clearly, if the threshold is set to 0, we will get one user categories. The complexity of clustering the users increases exponentially with the number of the user categories. In order to reduce the computational complexity of clustering the users, we have to choose an appropriate threshold. For that 2560 users data, we made several experiments to cluster that users data by constantly adjusting the value of the threshold. We can get seven user categories and get better distribution of user in Table 2 while the threshold was set to 0.6268. Certainly, it is allowed that the threshold is set higher or lower than 0.6268, but the distribution of user will become worse. Therefore, in order to achieve the purpose of the clustering users, we have to choose an appropriate clustering threshold. After making several experiments, we set the clustering threshold $\lambda = 0.6268$ in this paper and then divide 2560 users into seven categories in Table 2.

After the users clustering, we use formula (7) based on the cosine similarity to calculate the similarity between the target user and the other users. We can find out the nearest neighbor users of the target user according to the similarity value. In the similarity calculation, here there are two cases: (1) the target user belongs to the clustered users; (2) the target user does not belong to the clustered users. For case (1), the target user and his most nearest neighbors must belong to the same category of the clustered users; we only need to find out the k most nearest neighbors in the same category according to the value of the fuzzy similarity matrix. For case (2), we calculate the similarity value between the target user and the $\log_2^n$ clustered users at most based on the Binary Search Algorithm [28], where n is the number of clustered users. In order to reduce the amount of calculation of the recommendation actions after that, we add the target user into the same category with the most clustered user. In order to make the experiment results more objective and realistic, we choose an investor user that does not belong to the clustered users as the study target in the experiment and the result of the nearest seven neighbors of the target user in Table 3.

With the complement classified for the target user, we can get the k clustered users and get the stock set from those users. Here we call stock set as a prerecommended stock set. We use a method based on M/G/1 to compute the net inflow amount of big order for every stock in the prerecommended stock set. From the prerecommended stock set descending ordered by the value of big order net inflow, we choose the k stocks with highest value in front of the stock set as the last recommendation stock set for the target user. Due to limited paper space, we selected five stocks in each period of Figure 3 as the research stock in Tables 4, 5, 6, and 7. If the value of any stock in the stock set is smaller than 0, we have to set that the value of k is bigger and repeat the previous calculation steps. If the prerecommended stock set contains the all stock of the China Stock Market and any stock value of the big order net inflow is smaller than 0, we do not recommend any stock to the target user and recommend the target user to keep a wait state with holding money.

We do some research on some stocks that are recommended by the proposed model to analyze the maximum possible stocks return during the four different day intervals, that is 10 days, 30 days, 90 days, and 1 year, since that day when we recommend those stocks to the target user. By observing the experiment results, we found that those recommended stocks have better gains in Table 8, especially those stocks whose big order net inflow is higher proportion of trading volume of those days than the others. In addition, the stock price is the highest stock price without ex-rights or ex-dividend at the last trading day of the four corresponding day intervals.

Table 4: The recommended five stocks in the A period.

Stock name (code)	Date/mean	Buy	Sell	Net amount (10^4 share)	Percent (%)	Net money (10^4 yuan)
Jianghuai Automobile (600418)	2012-11-28	65.87	162.00	−96.11	−19.09	−481.40
	2012-11-29	77.98	22.48	55.50	8.52	283.60
	2012-11-30	476.20	216.60	259.50	21.37	1359.30
	2012-12-03	965.30	450.10	515.20	18.93	2801.70
	2012-12-04	1098.80	505.80	593.00	24.59	3285.90
	Mean	**536.83**	**271.40**	**265.42**	**10.86**	**1449.82**
Janus Precision (300083)	2012-11-28	0.00	0.00	0.00	0.00	0.00
	2012-11-29	6.49	0.00	6.49	17.22	68.47
	2012-11-30	0.00	0.00	0.00	0.00	0.00
	2012-12-03	3.35	0.00	3.35	12.75	35.17
	2012-12-04	3.26	4.00	−0.74	−1.54	−7.34
	Mean	**2.62**	**0.80**	**1.82**	**5.69**	**19.26**
Great Wall Motors (601633)	2012-11-28	58.62	51.76	6.86	1.21	123.00
	2012-11-29	93.53	109.20	−15.70	−4.20	−279.10
	2012-11-30	125.40	99.61	25.77	5.72	471.40
	2012-12-03	259.20	216.80	42.35	7.60	780.50
	2012-12-04	301.40	236.00	65.43	13.03	1196.20
	Mean	**167.63**	**142.67**	**24.94**	**4.67**	**458.40**
Huasun Group (000790)	2012-11-28	71.27	42.75	28.52	8.24	179.00
	2012-11-29	0.00	10.34	−10.34	−4.72	−62.21
	2012-11-30	6.62	0.00	6.62	2.84	40.48
	2012-12-03	21.60	23.88	−2.28	−0.81	−14.04
	2012-12-04	24.26	13.11	11.15	4.54	63.76
	Mean	**24.75**	**18.02**	**6.73**	**2.02**	**41.40**
Faw Car (000800)	2012-11-28	9.79	41.20	−31.41	−10.68	−184.70
	2012-11-29	23.03	24.87	−1.84	−1.64	−18.10
	2012-11-30	30.49	40.12	−9.64	−2.57	−56.06
	2012-12-03	241.80	245.50	−3.73	−0.43	−16.62
	2012-12-04	314.90	141.30	173.60	17.21	1075.10
	Mean	**124.00**	**98.60**	**25.40**	**0.38**	**159.92**

4.3. Experimental Results Analysis. In this section, we briefly analyze and explain the experimental results. After selecting an investor user that does not belong to the clustered users as the study target user in the experiment, we calculate the similarity value between the target user and the clustered users using formula (7) based on the cosine similarity. Because the nearest neighbors of the target user all belong to category 3 of clustered users, the target user belongs to category 3 according to the nature of the clustering that the target user and his nearest neighbors belong to the same cluster. We select the nearest seven neighbors of the target user to show you in Table 3. We use a method based on M/G/1 to compute the net inflow amount of big order for every stock in the prerecommended stock set selected from those users. We select five stocks with higher value of big order net inflow during the period between 2012/11/28 and 2012/12/04 as the recommended stocks for the target user in Table 4. For simplicity, we sort the stocks using big order net inflow daily mean ratio from large to small and select five stocks to recommend the target user.

Table 4 shows the daily big order net flow, ratio, and money for the five recommended stocks. In order to ensure the data neat and objective, we calculate the daily big order net flow for every stock at continuous five trading days and compute the average of them. Combining the big order net money inflow, we use the big order net inflow ratio as first evaluation criterion for the stocks. From the data in Table 3, we find that the number of days of big order net money inflow is generally greater than the number of days of big order net money outflow. Although there are four days of big order net money outflow in the stock of Faw Car, the big order net money inflow of the fifth day is greater than the sum of big order net money outflow of the previous four days. For that reason we believe that the stock will rise in the future. Generally, the big order net inflow only affects the short-term price of stock and shows that some investors are upbeat about the stock in the near future. The short-term price of stock may be consistent with how much the big order net money inflow is, but the long-term price of stock may not be fully consistent with it. As we all know, the long-term price of stock

Table 5: The recommended five stocks in the B period.

Stock name (code)	Date/mean	Buy	Sell	Net amount (10^4 share)	Percent (%)	Net money (10^4 yuan)
Dahu Aquaculture (600257)	2012-12-17	1765.20	1012.20	753.00	16.59	5306.90
	2012-12-18	1018.30	832.50	185.90	5.12	1280.60
	2012-12-19	1207.50	723.90	483.60	19.88	3346.80
	2012-12-20	1394.20	852.60	541.60	20.45	3766.90
	2012-12-21	597.40	463.50	133.90	6.17	956.00
	Mean	**1196.52**	**776.94**	**419.60**	**13.64**	**2931.44**
Changan Automobile (000625)	2012-12-17	432.20	300.40	131.80	7.11	720.60
	2012-12-18	427.10	840.50	−413.40	−17.84	−2185.60
	2012-12-19	5272.20	2329.90	2942.30	31.97	17000.00
	2012-12-20	2238.60	1906.90	331.70	6.30	1963.60
	2012-12-21	4297.70	2689.30	1608.40	22.43	10000.00
	Mean	**2533.56**	**1613.40**	**920.16**	**9.99**	**5499.72**
Xinning Modern Logistics (300013)	2012-12-17	246.70	101.70	144.90	43.29	1243.50
	2012-12-18	153.90	151.50	2.41	0.28	21.33
	2012-12-19	20.56	29.87	−9.32	−2.30	−80.77
	2012-12-20	9.09	13.62	−4.53	−0.96	−40.18
	2012-12-21	10.73	8.35	2.39	0.43	23.55
	Mean	**88.20**	**61.01**	**27.17**	**8.15**	**233.49**
Robam Appliances (002508)	2012-12-17	141.70	79.90	61.83	16.93	1150.70
	2012-12-18	12.43	24.04	−11.61	−9.03	−212.80
	2012-12-19	14.14	3.90	10.24	15.52	189.80
	2012-12-20	1.70	14.35	−12.65	−13.01	−228.30
	2012-12-21	79.50	76.40	3.09	1.08	60.14
	Mean	**49.89**	**39.72**	**10.18**	**2.30**	**191.91**
Great Wall Motor (601633)	2012-12-17	139.40	147.70	−8.27	−1.62	−133.00
	2012-12-18	125.70	136.20	−10.52	−1.95	−218.80
	2012-12-19	97.63	55.14	42.50	10.17	893.70
	2012-12-20	253.70	179.60	74.04	14.04	1608.70
	2012-12-21	146.40	142.20	4.21	1.06	90.46
	Mean	**152.57**	**132.17**	**20.39**	**4.34**	**448.21**

may be consistent with company's development, company's profitability, stock trader's goal and skill, and so on. We can see the above phenomenon from Table 8. Correspondingly, we can find that similar characteristics exist in Tables 5, 6, and 7.

Table 8 shows the recommended stocks maximum possible gains during the four different day intervals. In order to conveniently compare with the recommended stocks, we put the corresponding data of Shanghai Composite Index on the bottom of the data of the recommended stocks. Due to a similar trend of CSI 800 and Shanghai Composite Index, we chose Shanghai Composite Index as CSI in Table 8. If the highest price of the stock or CSI is equivalent in two or more continuous periods, this shows that the price of the stock or CSI is smaller than the previous price of that one and shows that the stock has been in decline during the following period. For example, the highest price of the CSI of the "highest price within 90 days" is equal to the highest price of CSI of the "highest price within 1 year"; they all are 2444.80 and we find that the running range of CSI is the interval from C to D in Figure 3. From Table 8, we find that the maximum possible gains of most of the recommended stocks are bigger than the CSI. Without a doubt, the target investor cannot get the same gains with the maximum possible gains because the time of buying or selling stock is uncertain for the target investor. It is easy to understand that the recommended stocks can get better gains between A period and C period in Figure 3. But it is very difficult that the recommended stocks can get better gains between C period and D period or after D period since the CSI has been going down or fluctuant. They fully illustrate the effectiveness of the proposed model.

In order to show clearly the return of the recommended stocks, we choose the C and D period in Figure 3. For that is the start period and middle period of the CSI fall or fluctuant, the general stocks will fall or fluctuate with the CSI. If the recommended stocks can get better returns in the falling period, it clearly shows that the proposed method is excellent. We select two last stocks from the recommended five stocks in the C and D period. Figures 4 and 5 show the return comparing the two stocks with the CSI in the 20 weeks after recommending those stocks. Because the CSI market was closed for a holiday, Figure 4 shows only 18 trading weeks and

Table 6: The recommended five stocks in the C period.

Stock name (code)	Date/mean	Buy	Sell	Net amount (10^4 share)	Percent (%)	Net money (10^4 yuan)
Irtouch Systems (300282)	2013-02-04	27.81	22.55	5.27	9.26	78.49
	2013-02-05	6.39	5.54	0.85	2.73	12.77
	2013-02-06	24.82	25.77	−0.95	−1.43	−13.60
	2013-02-07	9.18	2.36	6.82	16.88	103.20
	2013-02-08	18.23	7.83	10.40	19.58	161.00
	Mean	**17.29**	**12.81**	**4.48**	**9.40**	**68.37**
Transinfo Technology (002373)	2013-02-04	15.78	19.01	−3.23	−5.41	−36.05
	2013-02-05	3.28	7.37	−4.09	−13.08	−47.06
	2013-02-06	19.39	5.84	13.55	23.65	159.30
	2013-02-07	47.76	39.35	8.41	8.94	99.27
	2013-02-08	31.28	24.18	7.10	9.52	85.24
	Mean	**23.50**	**19.15**	**4.35**	**4.72**	**52.14**
Microgate Technology (300319)	2013-02-04	15.84	23.28	−7.44	−11.44	−126.90
	2013-02-05	32.69	15.87	16.82	20.43	293.30
	2013-02-06	14.57	14.34	0.23	0.44	4.55
	2013-02-07	14.77	10.36	4.41	8.45	77.36
	2013-02-08	22.16	19.66	2.50	2.96	44.09
	Mean	**20.01**	**16.70**	**3.30**	**4.17**	**58.48**
Kingsun Optoelectronic (002638)	2013-02-04	317.20	309.40	7.84	1.08	84.20
	2013-02-05	251.90	244.50	7.39	1.21	77.51
	2013-02-06	223.00	232.10	−9.15	−1.84	−94.10
	2013-02-07	229.90	181.00	48.91	10.29	509.60
	2013-02-08	256.90	218.50	38.42	6.88	408.50
	Mean	**255.78**	**237.10**	**18.68**	**3.52**	**197.14**
Victory Precision (002426)	2013-02-04	23.48	20.45	3.03	2.85	19.94
	2013-02-05	25.84	31.34	−5.50	−4.97	−35.74
	2013-02-06	10.90	5.02	5.88	8.48	38.82
	2013-02-07	3.58	10.97	−7.39	−14.02	−47.94
	2013-02-08	104.50	51.13	53.40	24.99	358.40
	Mean	**33.66**	**23.78**	**9.88**	**3.47**	**66.70**

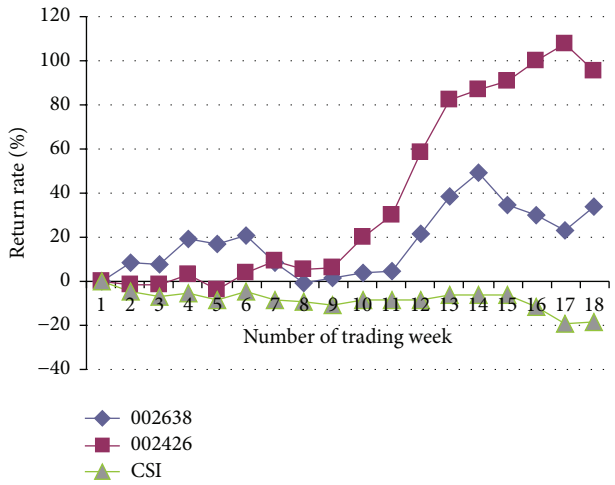


Figure 4: Return comparing of two stocks with the CSI in the C period.

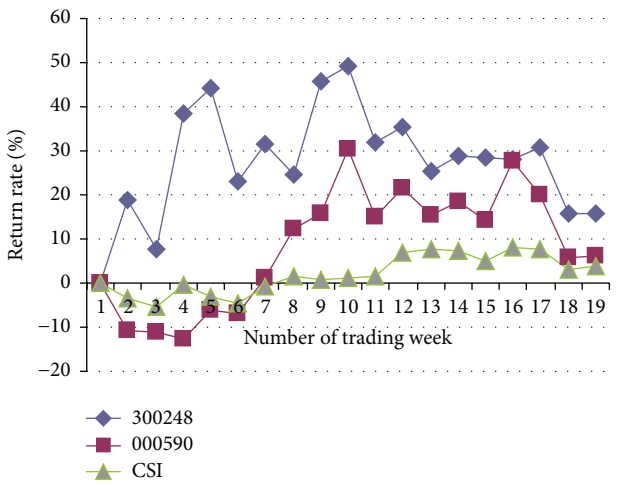


Figure 5: Return comparing of two stocks with the CSI in the D period.

Table 7: The recommended five stocks in the D period.

Stock name (code)	Date/mean	Buy	Sell	Net amount (10^4 share)	Percent (%)	Net money (10^4 yuan)
Sunway Communicate (300136)	2013-06-17	182.30	162.30	20.01	7.34	465.70
	2013-06-18	96.59	89.67	6.92	4.87	157.90
	2013-06-19	51.76	41.94	9.82	11.37	224.20
	2013-06-20	164.20	131.00	33.15	13.30	765.70
	2013-06-21	136.10	137.20	−1.18	−0.56	−15.57
	Mean	**126.19**	**112.42**	**13.74**	**7.26**	**319.59**
Huafon Spandex (002064)	2013-06-17	1097.00	693.20	403.90	21.44	2671.10
	2013-06-18	829.70	759.60	70.12	4.40	486.90
	2013-06-19	356.00	305.80	50.22	5.92	343.80
	2013-06-20	282.10	287.60	−5.49	−0.82	−38.74
	2013-06-21	168.40	151.50	16.88	3.33	114.30
	Mean	**546.64**	**439.54**	**107.13**	**6.85**	**715.47**
Tianshan Bio-Engineer (300313)	2013-06-17	122.80	115.90	6.90	2.38	96.30
	2013-06-18	195.50	170.60	24.88	6.01	351.50
	2013-06-19	156.70	143.20	13.49	3.83	199.30
	2013-06-20	104.60	106.10	−1.52	−0.64	−18.87
	2013-06-21	61.17	34.97	26.20	14.15	354.60
	Mean	**128.15**	**114.15**	**13.99**	**5.15**	**196.57**
Newcapec (300248)	2013-06-17	43.79	22.85	20.94	12.56	264.70
	2013-06-18	40.35	23.07	17.28	11.14	219.90
	2013-06-19	37.48	33.60	3.88	2.58	51.34
	2013-06-20	53.67	62.10	−8.43	−4.35	−100.70
	2013-06-21	39.32	46.01	−6.68	−4.83	−79.32
	Mean	**42.92**	**37.53**	**5.40**	**3.42**	**71.18**
Unisplendour Guhan (000590)	2013-06-17	134.40	149.50	−15.15	−1.45	−132.80
	2013-06-18	315.30	153.30	162.00	13.28	1677.50
	2013-06-19	62.76	62.54	0.22	0.04	5.04
	2013-06-20	84.21	63.16	21.05	4.03	212.60
	2013-06-21	29.58	65.09	−35.51	−6.31	−353.50
	Mean	**125.25**	**98.72**	**26.52**	**1.92**	**281.77**

Figure 5 shows only 19 trading weeks. Observing the return of the two stocks during the 20 weeks and comparing with the CSI over the same period, the two stocks get the better returns and it is far better than the CSI. The stock of Victory Precision (002426) gets the 107.96% best returns at the 17th trading week in Figure 4 and the stock of Newcapec (300248) gets the 49.21% best returns at the 10th trading week in Figure 5. Certainly, we cannot get the best return in practice, but the experienced investors will get far better returns than the CSI by purchasing the recommended stocks.

Figure 6 shows that different stock recommendation models will bring different stock returns. We compute the mean return rate of 18 trading weeks for 20 stocks in two different stocks sets which were recommended by the proposed method and the traditional method in the above four different periods and Figure 6 shows the result. Assuming that the target user bought the frontal stocks in the recommend stocks set, we find that the target user with appropriate operating will get far better stock return rate as the top line indicated in Figure 6.

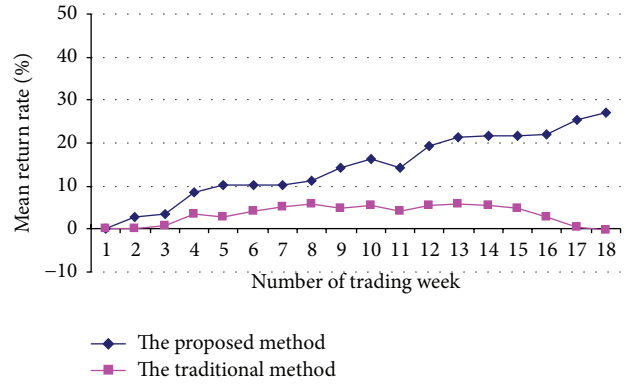


Figure 6: Mean return rate comparing of two different methods.

In Section 4.1, we selected data at four different periods of the real stock market in China. Each of the periods is very small and has only five trading days. What is the reason for validating the recommended stocks in experimentation

Table 8: The recommended stocks maximum possible gains during the four different day intervals.

Recommend date/period	Stock name (code)/CSI	Recommend buy/open Price	Highest price within 10 days	Rise-up within 10 days	Highest price within 30 days	Rise-up within 30 days	Highest price within 90 days	Rise-up within 90 days	Highest price within 1 year	Rise-up within 1 year
2012-12-05 A period	Jianghuai Automobile (600418)	5.56	6.01	8.09%	6.98	25.54%	9.39	68.88%	10.37	86.51%
	Janus Precision (300083)	10.14	11.60	14.40%	12.55	23.77%	17.35	71.10%	25.00	146.55%
	Great Wall Motor (601633)	18.43	20.80	12.86%	23.91	29.73%	31.50	70.92%	52.85	186.76%
	Huasun Group (000790)	5.85	6.67	14.02%	7.10	21.37%	8.55	46.15%	12.89	120.34%
	Faw Car (000800)	6.30	7.13	13.17%	8.72	38.41%	8.79	39.52%	15.88	152.06%
	Shanghai Composite Index (CSI)	**1973.11**	**2152.50**	**9.09%**	**2296.11**	**16.37%**	**2444.80**	**23.91%**	**2444.80**	**23.91%**
2012-12-24 B period	Dahu Aquaculture (600257)	6.83	7.42	8.64%	7.42	8.64%	7.42	8.64%	9.36	37.04%
	Changan Automobile (000625)	6.17	6.73	9.08%	7.62	23.50%	9.56	54.94%	13.19	113.78%
	Xinning Modern (300013)	9.20	11.43	24.24%	11.43	24.24%	11.43	24.24%	19.97	117.07%
	Robam Appliances (002508)	18.62	19.35	3.92%	25.00	34.26%	26.87	44.31%	42.79	129.81%
	Great Wall Motor (601633)	22.45	23.91	6.50%	30.39	35.37%	34.18	52.25%	52.85	135.41%
	Shanghai Composite Index (CSI)	**2150.65**	**2296.11**	**6.76%**	**2335.81**	**8.61%**	**2444.80**	**13.68%**	**2444.80**	**13.68%**
2013-02-18 C period	Irtouch Systems (300282)	15.55	15.58	0.19%	16.61	6.82%	16.61	6.82%	32.69	110.23%
	Transinfo Technology (002373)	11.90	12.22	2.69%	12.73	6.97%	14.01	17.73%	19.50	63.87%
	Microgate Technology (300319)	17.65	18.62	5.50%	18.66	5.72%	18.66	5.72%	37.63	113.20%
	Kingsun Optoelectronic (002638)	10.80	12.65	17.13%	13.19	22.13%	15.42	42.78%	16.20	50.00%
	Victory Precision (002426)	6.78	7.22	6.49%	7.22	6.49%	12.50	84.37%	20.46	201.77%
	Shanghai Composite Index (CSI)	**2444.80**	**2444.80**	**0.00%**	**2444.80**	**0.00%**	**2444.80**	**0.00%**	**2444.80**	**0.00%**
2013-06-24 D period	Sunway Communicate (300136)	23.05	27.20	18.00%	27.20	18.00%	27.20	18.00%	27.66	20.00%
	Huafon Spandex (002064)	6.64	7.73	16.42%	8.30	25.00%	11.50	73.19%	11.50	73.19%
	Tianshan Bio-Engineer (300313)	13.41	14.44	7.68%	14.44	7.68%	16.63	24.01%	23.10	72.26%
	Newcapec (300248)	11.99	14.85	23.85%	17.90	49.29%	19.19	60.05%	31.32	161.22%
	Unisplendour Guhan (000590)	9.90	9.38	−5.25%	9.39	−5.15%	13.50	36.36%	17.79	79.70%
	Shanghai Composite Index (CSI)	**2068.86**	**2068.86**	**0.00%**	**2092.87**	**1.16%**	**2270.27**	**9.74%**	**2270.27**	**9.74%**

on larger period with one to eighteen weeks in Figures 4 and 5? As been well known, it is easy for buying stocks with big order, but it is very difficult for selling stocks with big order. The investors which can buy or sell stocks are generally institutional investors. The institutional investors mainly refer to a number of financial institutions, including banks, insurance companies, investment trust companies, credit unions, national bodies to establish a pension fund, and other organizations.

It is very difficult or impossible that the institutional investors can sell out the stocks within a few days after buying lots of stocks with big order. There may be two cases. The first case is that the institutional investors want to sell out all stocks in the next period of day. In general, selling out all the stocks required time is two to four weeks depending on the amount of their holding stocks which is generally more than the amount of buying stocks in those days we select data. The other case is that the institutional investors want to buy and not to sell some stocks in the next period of day. For example, this day when we select data is the starting period of buying stocks for the institutional investors. After some days, the amount of their buying stocks reached their desired amount and the institutional investors are no longer buying stocks. They want to sell out all the stocks when the stocks reach a certain range of price. In general, the time is twelve to sixteen weeks from buying stocks to selling out stocks for the institutional investors. In fact, nobody knows when they buy and when to sell out. In order to validate the proposal, we make the experimentation at the large period from one to eighteen weeks after our recommend date. As we see in Figures 4 and 5, the stock of Victory Precision (002426) and the stock of Kingsun Optoelectronic (002426) get, respectively, the best returns at the 17th trading week and at the 14th trading week in Figure 4 and the stock of Newcapec (300248) gets the better returns at the 4th trading week and it gets the best returns at the 10th trading week in Figure 5. Certainly, we cannot get the best return in practice, because nobody knows the highest price of the stocks in certain period and when they ought to sell out the stocks.

5. Conclusions and Future Works

In this paper, we proposed an efficient stock recommendation model based on big order net inflow. The proposed stock recommendation model based on big order net inflow can not only filter some low investment value stocks and improve the prediction accuracy in order to recommend some more investment value stocks to the target users and get more stock returns, but also improve the speed of compute by using some advanced algorithms, such as Binary Search Algorithm, and satisfy the investment desire of the investors in real life. From the experimental results, we found that the proposed stock recommendation model has a better performance than the model based on the money flow, and then it can filter the low or negative investment returns stocks and improve the investment gains for target investors.

In the future, the next work is to study how to improve the accuracy of recommendation model. We will reduce the impact of the model on the influence of corporate events and rumors of trader and apply the model to more stock markets or future markets in the different country. We think that there are three aspects worth studying. Firstly, we can improve model performance with machine learning algorithms, time series analysis or latest technology of fuzzy cluster, and so on. For example, we may improve the accuracy thought training a large number of historical stocks data sets from different markets and countries using the machine learning algorithms. Secondly, we may improve the accuracy of the clustering set for stocks or users through selecting appropriate attributes of them or adding weights for every attribute. Finally, we may study how to find the big order transactions in real trading process, because some institutional investors may subdivide the big order into many random small orders by quantitative trading system software; it is very difficult for us.

Conflict of Interests

All the authors of this paper declare that they have no conflict of interests in connection with the work submitted.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (no. 61370073), the National High Technology Research and Development Program of China (no. 2007AA01Z423), Sichuan Province Science and Technology Support Program (no. 2013GZX0165), the Constructing Program of the Key Discipline in Huaihua University, the Scientific Research Fund of Hunan Provincial Education (nos. 12C0840 and 14C0886), the Scientific Research Fund of Huaihua University (nos. HHUY2012-15 and HHUY2011-17), the Science and Technology Plan Projects of Huaihua City, and the Key Laboratory of Intelligent Control Technology for Wuling-Mountain Ecological Agriculture in Hunan Province (no. ZNKZ2014-9).

References

[1] J. Duan, H. Liu, and J. Zeng, "Posterior probability model for stock return prediction based on analyst's recommendation behavior," *Knowledge-Based Systems*, vol. 50, pp. 151–158, 2013.

[2] E. J. de Fortuny, T. de Smedt, D. Martens, and W. Daelemans, "Evaluating and understanding text-based stock price prediction models," *Information Processing & Management*, vol. 50, no. 2, pp. 426–441, 2014.

[3] L. Huo, B. Jiang, T. Ning, and B. Yin, "A BP neural network predictor model for stock price," in *Intelligent Computing Methodologies*, D. S. Huang, K. H. Jo, and L. Wang, Eds., Lecture Notes in Computer Science, pp. 362–368, Springer, Berlin, Germany, 2014.

[4] A. A. Adebiyi, A. O. Adewumi, and C. K. Ayo, "Comparison of ARIMA and artificial neural networks models for stock price prediction," *Journal of Applied Mathematics*, vol. 2014, Article ID 614342, 7 pages, 2014.

[5] K. Miwa and K. Ueda, "Slow price reactions to analysts' recommendation revisions," *Quantitative Finance*, vol. 14, no. 6, pp. 993–1004, 2014.

[6] T. Geva and J. Zahavi, "Empirical evaluation of an automated intraday stock recommendation system incorporating both market data and textual news," *Decision Support Systems*, vol. 57, no. 1, pp. 212–223, 2014.

[7] C. S. Han and C. Y. Wan, "Future stock price predicting system for use in enterprise, has future prediction compute server setting up weighted value and producing virtual total amount of market price information based on basis predictive model," KR2014120416-A, to Cs Co Ltd, Korea Advanced Institute of Science & Technology, 2014.

[8] N. C. Brown, K. D. Wei, and R. Wermers, "Analyst recommendations, mutual fund herding, and overreaction in stock prices," *Management Science*, vol. 60, no. 1, pp. 1–20, 2014.

[9] M.-Y. Chen, "A high-order fuzzy time series forecasting model for internet stock trading," *Future Generation Computer Systems*, vol. 37, pp. 461–467, 2014.

[10] B. Q. Sun, H. F. Guo, H. R. Karimi, Y. Ge, and S. Xiong, "Prediction of stock index futures prices based on fuzzy sets and multivariate fuzzy time series," *Neurocomputing*, vol. 151, no. 3, pp. 1528–1536, 2015.

[11] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques," *Expert Systems with Applications*, vol. 42, no. 1, pp. 259–268, 2015.

[12] D. Sonsino and T. Shavit, "Return prediction and stock selection from unidentified historical data," *Quantitative Finance*, vol. 14, no. 4, pp. 641–655, 2014.

[13] M.-Y. Chen and B.-T. Chen, "A hybrid fuzzy time series model based on granular computing for stock price forecasting," *Information Sciences*, vol. 294, pp. 227–241, 2015.

[14] Z. Xin, Z. Ma, and M. Gu, "Fuzzy clustering collaborative recommendation algorithms served for directional information recommendation," *Computer Science*, vol. 34, no. 9, pp. 128–130, 2007.

[15] L. Chapple and J. E. Humphrey, "Does board gender diversity have a financial impact? Evidence using stock portfolio performance," *Journal of Business Ethics*, vol. 122, no. 4, pp. 709–723, 2014.

[16] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[17] Q. Xu, "Continuous time M/G/1 queue with multiple vacations and server close-down time," *Journal of Computational Information Systems*, vol. 3, no. 2, pp. 753–757, 2007.

[18] A. Frazzini and O. A. Lamont, "Dumb money: mutual fund flows and the cross-section of stock returns," *Journal of Financial Economics*, vol. 88, no. 2, pp. 299–322, 2008.

[19] I. Gath and E. Bar-On, "Computerized method for scoring of polygraphic sleep recordings," *Computer Programs in Biomedicine*, vol. 11, no. 3, pp. 217–223, 1980.

[20] M. Hong-Wei, Z. Guang-Wei, and L. Peng, "Survey of collaborative filtering algorithms," *Mini-Micro Systems*, vol. 7, pp. 1282–1288, 2009.

[21] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International World Wide Web Conference*, pp. 285–295, May 2001.

[22] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[23] 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.

[24] G. Choudhury, "Some aspects of M/G/1 queue with two different vacation times under multiple vacation policy," *Stochastic Analysis and Applications*, vol. 20, no. 5, pp. 901–909, 2002.

[25] B. T. Doshi, "Conditional and unconditional distributions for M/G/1 type queues with server vacations," *Questa*, vol. 7, pp. 229–252, 1990.

[26] H. W. Lee, "M/G/1 queue with exceptional first vacation," *Computers & Operations Research*, vol. 15, no. 5, pp. 441–445, 1988.

[27] K. K. Leung, "On the additional delay in an M/G/1 queue with generalized vacations and exhaustive service," *Operations Research*, vol. 40, supplement 2, pp. S272–S283, 1992.

[28] A. Hatamlou, "In search of optimal centroids on data clustering using a binary search algorithm," *Pattern Recognition Letters*, vol. 33, no. 13, pp. 1756–1760, 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 683176, 9 pages
http://dx.doi.org/10.1155/2015/683176

Research Article

Image Segmentation by Edge Partitioning over a Nonsubmodular Markov Random Field

Ho Yub Jung[1] and Kyoung Mu Lee[2]

[1]*Division of Computer and Electronic Systems Engineering, Hankuk University of Foreign Studies, Yongin 449-791, Republic of Korea*
[2]*Department of Electrical and Computer Engineering, College of Engineering, Seoul National University, Seoul 151-744, Republic of Korea*

Correspondence should be addressed to Ho Yub Jung; jung.ho.yub@gmail.com

Received 26 July 2015; Revised 16 November 2015; Accepted 3 December 2015

Academic Editor: Costas Panagiotakis

Edge weight-based segmentation methods, such as normalized cut or minimum cut, require a partition number specification for their energy formulation. The number of partitions plays an important role in the segmentation overall quality. However, finding a suitable partition number is a nontrivial problem, and the numbers are ordinarily manually assigned. This is an aspect of the general partition problem, where finding the partition number is an important and difficult issue. In this paper, the edge weights instead of the pixels are partitioned to segment the images. By partitioning the edge weights into two disjoints sets, that is, cut and connect, an image can be partitioned into all possible disjointed segments. The proposed energy function is independent of the number of segments. The energy is minimized by iterating the QPBO-α-expansion algorithm over the pairwise Markov random field and the mean estimation of the cut and connected edges. Experiments using the Berkeley database show that the proposed segmentation method can obtain equivalently accurate segmentation results without designating the segmentation numbers.

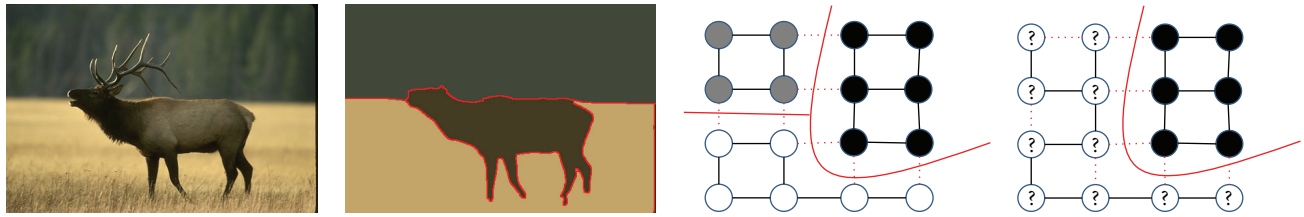
1. Introduction

There are numerous approaches and applications for unsupervised image segmentation in computer vision. Many different theories are proposed for varying the roles of the unsupervised segmentation. As a low level vision problem, an image can be simplified by oversegmentation using a number of different approaches, such as mode-seeking mean shift, multilevel thresholding, histogram-based neural networks, superpixel algorithms, and various graph-based methods [1–4]. Conversely, semantic segmentation is attempted for simultaneous detection, recognition, and segmentation [5].

Generally, the role of unsupervised segmentation falls between image simplification and full semantic segmentation, where semantically meaningful segments are expected to be found but not necessarily recognized. Segmentation is posed as an image-coloring problem that minimizes specific energy functions. Energy functions can be optimized using stochastic methods such as deterministic annealing and stochastic clustering [6–10]. For graph theoretic segmentation approaches, the spectral method and graph cut are efficient deterministic optimization methods [11–13]. Another traditional segmentation method is the variational method, which evolves boundary contours in a level set framework [14, 15].

The edge weight-based segmentation methods have evolved together with graph partition problems. When edge weights are all positive, the minimum cut can be found; however, the minimum cut has bias toward smaller cuts. Adding negative edge weights can prevent the problem so the graph becomes nonsubmodular; however, the problem becomes NP-hard [16]. Different algorithms have been introduced to estimate the correlation in clustering problem [17, 18]. In contrast, Shi and Malik normalized nonnegative edge weights so the bias toward smaller cuts was eliminated [11].

For the graph theoretic segmentation and level set methods, the number of segments must be predefined. The segment number choice greatly influences the quality of



(a) (b) (c) (d)

Figure 1: An image can be segmented by partitioning edges into two sets. Cut (dotted red) and connected (solid black) edge sets can be translated into a unique segmentation as in (c). However, it is also possible to have edge partitions that contradict the label assignments as in (d). By finding the image labeling that minimizes the edge partition energy, edge partitions like (d) are prevented, and a consistent image segmentation becomes possible as shown in (a) and (b).

segmentation, especially for a normalized cut. Nonetheless, there have been attempts to solve this problem. The number of segments can be controlled by setting the threshold value to the recursive normalized cut [11]. For level set approaches, a four-color theorem was used to segment images with an arbitrary number of phases with one or two level set functions [19]. However, these methods are still functions of K, the number of segments.

In this paper, transforming the pixel clustering problem into an edge partition problem circumvents the segment number selection problem. Edges among adjacent pixels can represent dissimilarity or similarity weights. Two edge partitions are always sufficient for pixel-partitioning problems. An edge can be in a cut set or connected set, which can then be translated into a unique segmentation, as in Figure 1(c). The cut edges indicate that the two node labels are different, whereas the connected edges indicate that two nodes have the same labels. In most cases, however, the cut or connect assignments on the edges are not enough to define a specific segmentation configuration, as in Figure 1(d). Random cut and connect assignments on the edges may result in contradiction of the node labels. However, under the pixel coloring framework, cut and connect assignments on the edges are defined concurrently with pixel labels, and inconsistencies, such as those in Figure 1(d), are prevented.

Under the pixel-labeling framework, a label number selection problem arises. Although the label number selection might seem similar to the segment number selection problem, there are subtle differences. First, pixels do not need to use all label assignments; thus, low numbers of segments are possible with large numbers of labels. Second, under the four-color map theorem, the maximum number of labels for two-dimensional (2D) segmentation can be as low as four. The four-color map theorem states that any 2D map can be colored with intact borders using a maximum of four colors [26]. This theorem can be translated directly to the segmentation problem; any 2D image segmentation can be represented using four labels [19].

In the following sections, a new energy function is introduced for image segmentation through the edge partition. The edge partitions can uniquely define the image segmentation with the hard constraints enforced by the image-labeling framework. Next, an energy minimization algorithm is proposed for the edge partitioning. The experimental section discusses tests of the proposed algorithm using the Berkeley image segmentation database.

2. Pixel Clustering

Image segmentation can be viewed as a pixel-partitioning problem. Many image segmentation methods borrow their ideas from the general partitioning techniques. The K-means algorithm minimizes the following function and segments the image into K regions. Consider

$$E_k = \sum_{i=1}^{k} \sum_{w_j \in S_i} \left\| w_j - \mu_i \right\|^2, \tag{1}$$

where w_j is the pixel feature value and μ_i is the mean values of the respective partition S_i. The K-means algorithm minimizes the sum of the squared distance from the mean of each partition. The energy function must have a fixed segmentation number. However, estimating the number of segments is a difficult task, and the number of partitions is often designated by human discretion.

3. Edge Partitions

An image can be represented as a set of nodes and edges by a graph $G = (\mathcal{V}, \mathcal{E})$. An edge $(u, v) \in \mathcal{E}$ is assigned with weight w between nodes $u, v \in \mathcal{V}$. For each node $v \in \mathcal{V}$, a label from $x_v \in \{1, 2 \cdots\}$ is assigned to define a segmentation.

3.1. Energy Function. The segmentation problem is formulated in terms of edge partitions. The edges can be partitioned into two sets C_{off} (cut) and C_{on} (connect), such that $C_{\text{off}} \cup C_{\text{on}} = \mathcal{E}$ and $C_{\text{off}} \cap C_{\text{on}} = \emptyset$. If an edge is in C_{on}, the pixel nodes connected by the edge have the same label. Otherwise, if an edge is in C_{off}, the pixel nodes connected by the edge have a different label

$$\begin{aligned} C_{\text{off}} &= \{w(u, v) \in \mathcal{E} \text{ s.t } x_u \neq x_v\}, \\ C_{\text{on}} &= \{w(u, v) \in \mathcal{E} \text{ s.t } x_u = x_v\}. \end{aligned} \tag{2}$$

In (2), $w(u, v)$ is an edge between pixel nodes u and v. The pixel labels for pixels u and v are denoted by x_u and x_v, respectively. $w(u, v)$ is a positive edge weight between pixel

nodes u and v. $w(u,v)$ can be a similar or dissimilar measure between the two pixel nodes. A simple example of $w(u,v)$ is the absolute difference between two pixel colors. Thus, if the colors between the two pixels have a large difference, the edge will likely be in C_{off}. If the two pixel colors have a small difference, the edge should be in C_{on}. The mean edge weight values of the C_{on} and C_{off} edge sets are found in the following equation:

$$\mu_{C_{\text{off}}} = \frac{\sum_{(u,v)\in C_{\text{off}}} w(u,v)}{|C_{\text{off}}|}, \qquad \mu_{C_{\text{on}}} = \frac{\sum_{(u,v)\in C_{\text{on}}} w(u,v)}{|C_{\text{on}}|}, \tag{3}$$

and then the energy function associated with the edge partitions can be defined by following equation:

$$E = \sum_{(u,v)\in C_{\text{off}}} \left(w(u,v) - \mu_{C_{\text{off}}}\right)^2 + \sum_{(u,v)\in C_{\text{on}}} \left(w(u,v) - \mu_{C_{\text{on}}}\right)^2. \tag{4}$$

$|\cdot|$ is the cardinality of the set. The energy function (4) is the same as the K-means algorithm in (1) except that the number of partitions is set to $K = 2$. The proposed energy function has two mean centers, but it also has hard constraints in (2). Regardless of the segmentation number, there can only be two partitions for the edges, cut C_{off} and connected edges C_{on}.

The proposed energy function breaks down into an image-labeling problem in order to maintain the label consistency conditions of (2). The image label state $\mathbf{x} = (x_1, x_2, \ldots, x_{|\mathscr{V}|})$ that minimizes (4) under (3) and (2) constraints is the proposed segmentation state. The number of labels must be at least two to avoid division by zero in (3). Under the well-known four-color map theorem, four labels $\{1, 2, 3, 4\}$ are sufficient to define all possible segment configurations for 2D images [19].

3.2. Optimization. Given the image label state $\mathbf{x}$, the mean values $\mu_{C_{\text{off}}}, \mu_{C_{\text{on}}}$ can be estimated as in (3). Otherwise, if $\mu_{C_{\text{off}}}$ and $\mu_{C_{\text{on}}}$ are kept constant, the image label state $\mathbf{x}$ can be found by optimizing the following pairwise energy function:

$$E_L = \sum_{(u,v)\in\mathscr{E}} \theta_{uv}(x_u, x_v), \tag{5}$$

$$\theta_{uv}(x_u, x_v) = \left(w(u,v) - \mu_{C_{\text{off}}}\right)^2 \quad \text{if } x_u \neq x_v, \qquad \theta_{uv}(x_u, x_v) = \left(w(u,v) - \mu_{C_{\text{on}}}\right)^2 \quad \text{if } x_u = x_v. \tag{6}$$

If the labels between edges are not the same, the edge is considered to be in the C_{off} cut set; otherwise, it belongs to the C_{on} connected set. With $\mu_{C_{\text{off}}}$ and $\mu_{C_{\text{on}}}$ constants, minimizing (5) is equivalent to minimizing the edge partition function (4).

The multilabel pairwise energy function (5) can be solved by QPBO-α-expansion. QPBO-α-expansion optimizes the multilabel MRFs by iteratively expanding a single label using graph cut [27]. Graph cut can find the optimal expansion if the expansion is submodular. In this problem, the expansions are nonsubmodular. The pairwise potentials for QPBO-α-expansion, where $x_u, x_v \in \mathbf{x}$ is the current label state, can be defined as follows:

$$\begin{aligned}
&\theta_{uv}(x_u, x_v) = \left(w(u,v) - \mu_{C_{\text{off}}}\right)^2 \quad \text{if } x_u \neq x_v,\\
&\theta_{uv}(x_u, x_v) = \left(w(u,v) - \mu_{C_{\text{on}}}\right)^2 \quad \text{if } x_u = x_v,\\
&\theta_{uv}(x_u, \alpha_v) = \left(w(u,v) - \mu_{C_{\text{off}}}\right)^2 \quad \text{if } x_u \neq \alpha_v,\\
&\theta_{uv}(x_u, \alpha_v) = \left(w(u,v) - \mu_{C_{\text{on}}}\right)^2 \quad \text{if } x_u = \alpha_v,\\
&\theta_{uv}(\alpha_u, x_v) = \left(w(u,v) - \mu_{C_{\text{off}}}\right)^2 \quad \text{if } \alpha_u \neq x_v,\\
&\theta_{uv}(\alpha_u, x_v) = \left(w(u,v) - \mu_{C_{\text{on}}}\right)^2 \quad \text{if } \alpha_u = x_v,\\
&\theta_{uv}(\alpha_u, \alpha_v) = \left(w(u,v) - \mu_{C_{\text{on}}}\right)^2.
\end{aligned} \tag{7}$$

This nonsubmodular binary labeling problem can be approached using the QPBO algorithm [28] with the possibility of a large number of unlabeled nodes. Recently introduced, QPBO improve (QPBOI) algorithm can cope with unlabeled regions [28]; however, this algorithm is not as efficient as the graph cut which minimizes the submodular potentials. The QPBOI algorithm can randomly improve the solution, but iterations of the improved steps can be time-consuming for large numbers of nodes.

Similar to the original K-means algorithm, good initialization is helpful to the optimization. The initial estimation of the means, $\mu_{C_{\text{off}}}$ and $\mu_{C_{\text{on}}}$, can be found by a K-means algorithm minimization of edge partitions (4) without the labeling constraint of (2). To estimate the initial state $\mathbf{x}$, the pixel clustering K-means algorithm (1) can be used. The general framework is illustrated in Algorithm 1.

3.3. Edge Weights. Various examples of the edge partition segmentation results using the color distance edge weights are shown in Figure 2 for the MSRC image database [29]. The color distance from the neighboring pixels is sufficient for some image segmentation problems, but more rigorous weight calculations are often suited for semantic segmentation. Instead of proposing new edge weight calculations, an existing state-of-the-art contour detection algorithm is incorporated.

The global probability of the boundary (GPB) edge detection method [25, 30], which scored best for the Berkeley database (http://www.cs.berkeley.edu/projects/vision/bsds), is employed as the edge weights. The edge weights can be connected between the pixel nodes, and the proposed edge partitioning algorithm can be implemented. Figure 3 shows the other segmentation results under the pixel-to-pixel edge connections. Although Figures 3(a) and 3(b) show a good segmentation result, the QPBOI algorithm cannot obtain

(1) Estimate image label state $\mathbf{x}$ with the K-means algorithm on pixels, $K_p = 4$.
(2) Estimate $\mu_{C_{\text{off}}}$ and $\mu_{C_{\text{on}}}$ by the K-means algorithm minimization of (4) without the labeling constraint of (2).
(3) Estimate the image label state $\mathbf{x}$ using the QPBOI-α-expansion. (keep $\mu_{C_{\text{off}}}$ and $\mu_{C_{\text{on}}}$ constant)
(4) Estimate $\mu_{C_{\text{off}}}$ and $\mu_{C_{\text{on}}}$ from the image label state $\mathbf{x}$.
(5) If $\mu_{C_{\text{off}}}$ and $\mu_{C_{\text{on}}}$ are unchanged, terminate. Else, repeat steps 3 and 4.

Algorithm 1: Minimizing Edge Partition.

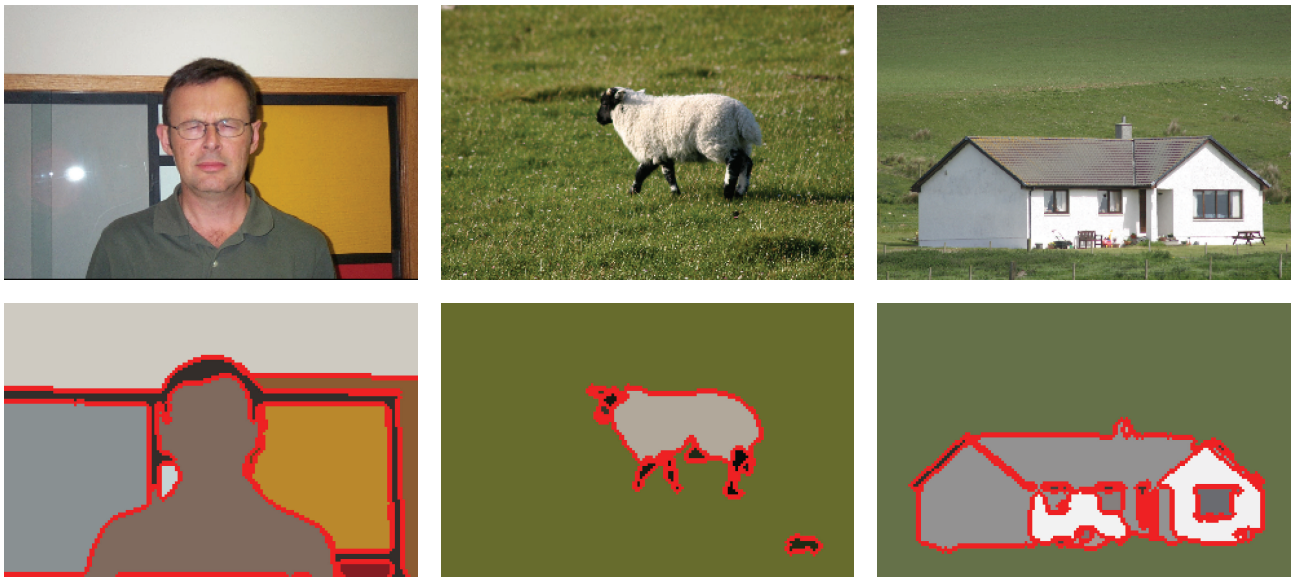


Figure 2: Even with the simple color distance weight, the edge partitions can produce adequate segmentation results. Some of the segmentation results from the MSRC database are shown.

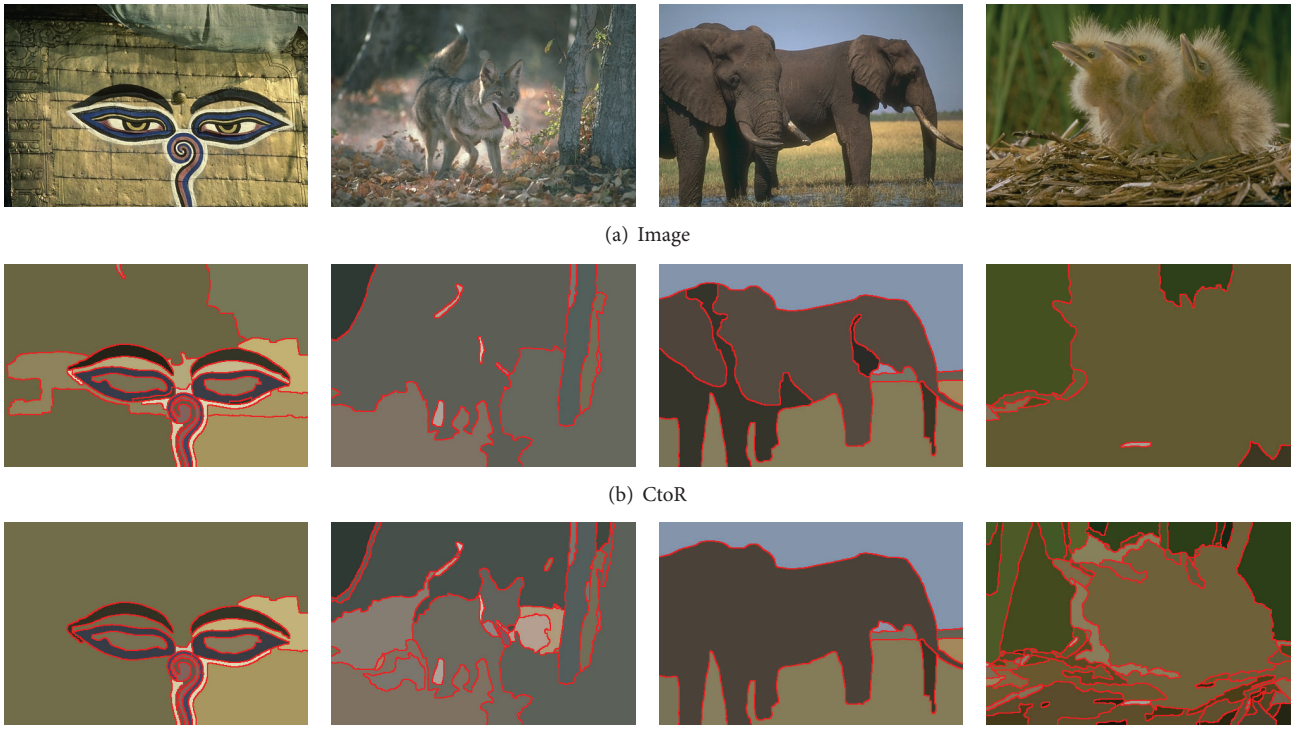
(a) (b) (c) (d)

Figure 3: The QPBOI optimization scheme can be efficient for (a) and (b). However, for many cases, the nonsubmodular potentials are too strong, and the QPBOI optimization fails in (c) and (d). More iterations of QPBOI can improve the result; however, iterations are time-consuming without improvement guarantees. Superpixel images are used in this study to reduce the computation time and increase the QPBOI iterations.

a good segmentation in Figures 3(c) and 3(d). The QPBOI algorithm often fails in the presence of a large number of nodes. Thus, to reduce both the computational time and the chance of failure in the QPBOI algorithm, the oversegmentation process is adopted from [25] in this segmentation. The edges are connected between the superpixels instead of the pixels. The number of oversegments is between 400 and 1000. The edge partitioning algorithm segments a BSDS image average in under 5 seconds.

4. Evaluation

The proposed edge partition approach is evaluated using the popular Berkeley image database. The set contains 300



(a) Image

(b) CtoR

(c) EPartition

Figure 4: The segmentation results from Table 1 are shown for CtoR and the proposed EPartition.

images with at least four human segment annotations per image. The three quantitative evaluation methods used are as follows: Probabilistic Rand Index (PRI) [31], Variation of Information (VoI) [32], and Boundary Displacement Error (BDE) [33]. Global Consistency Error (GCE) [34] is not included in this evaluation. GCE measures the extent to which one segmentation can be viewed as a refinement of another. However, one pixel per segment and one segment for an entire image can give zero error for GCE [31]. GCE favors extremely oversegmented or undersegmented results, and both cases are unwanted for a semantic segmentation. GCE is deemed to be an inconsistent evaluation method.

The evaluation methods used in this study are PRI, VoI, and BDE. PRI counts the number of consistent labels between the segmentation and the ground truth. VoI measures the segmentation randomness that cannot be explained by the ground truth. BDE is the average displacement error or the boundary pixels between two segmentation results. PRI counts the correctness in segmentation, while VoI and BDE measure the errors between the segmentation and ground truth. In the first subsection, the proposed method is evaluated against various segmentation methods. In the second subsection, the comparison between the proposed and the merge-threshold methods is demonstrated using the same edge weights.

4.1. Comparison to the Previous Segmentation Methods. Generally, the parameters are constant for the entire database and test methods. This evaluation includes mean shift (MShift) [1], graph-based segmentation (GBIS) [21], JSEG [20], Normalized Tree Partitioning (NTP) [22], saliency-based segmentation (Saliency) [23], Boundary Encoding Based Segmentation (TBES) [24], normalized cut (Ncut) [11], and fully connected spectral segmentation (SpecSeg) [13]. Additionally, contour to region (CtoR) [25] uses the same edge weights. Table 1 summarizes the performance of these methods. Many of the evaluation results are obtained from [13].

For PRI measurements, the merge-threshold method of CtoR ranks first. The proposed segmentation ranks first for VoI and BDE. The CtoR method is available to the public by the authors. The threshold value for the CtoR method was chosen to be 80 for its highest average ranking. A number of segmentation results of CtoR and of the proposed EPartition are shown in Figure 4. For the normalized cut and fully connected spectral segmentation, the segmentation number is chosen for each image and is excluded from the rankings.

CtoR and EPartition use the same edge weights; thus, their performances are similar. However, in CtoR, a merge-threshold algorithm is used for segmentation. Different thresholds among integer intervals $\{1, \ldots, 255\}$ are shown for the PRI, VoI, and BDE evaluation methods in Figure 5. Generally, PRI and BDE favor oversegmentation and VoI favors undersegmentation. The optimal threshold value is generally smaller for PRI and BDE than VoI.

In contrast, the edge partitioning segmentation is independent of a threshold value. Figure 5 shows the performance of the CtoR merge-threshold method in terms of threshold

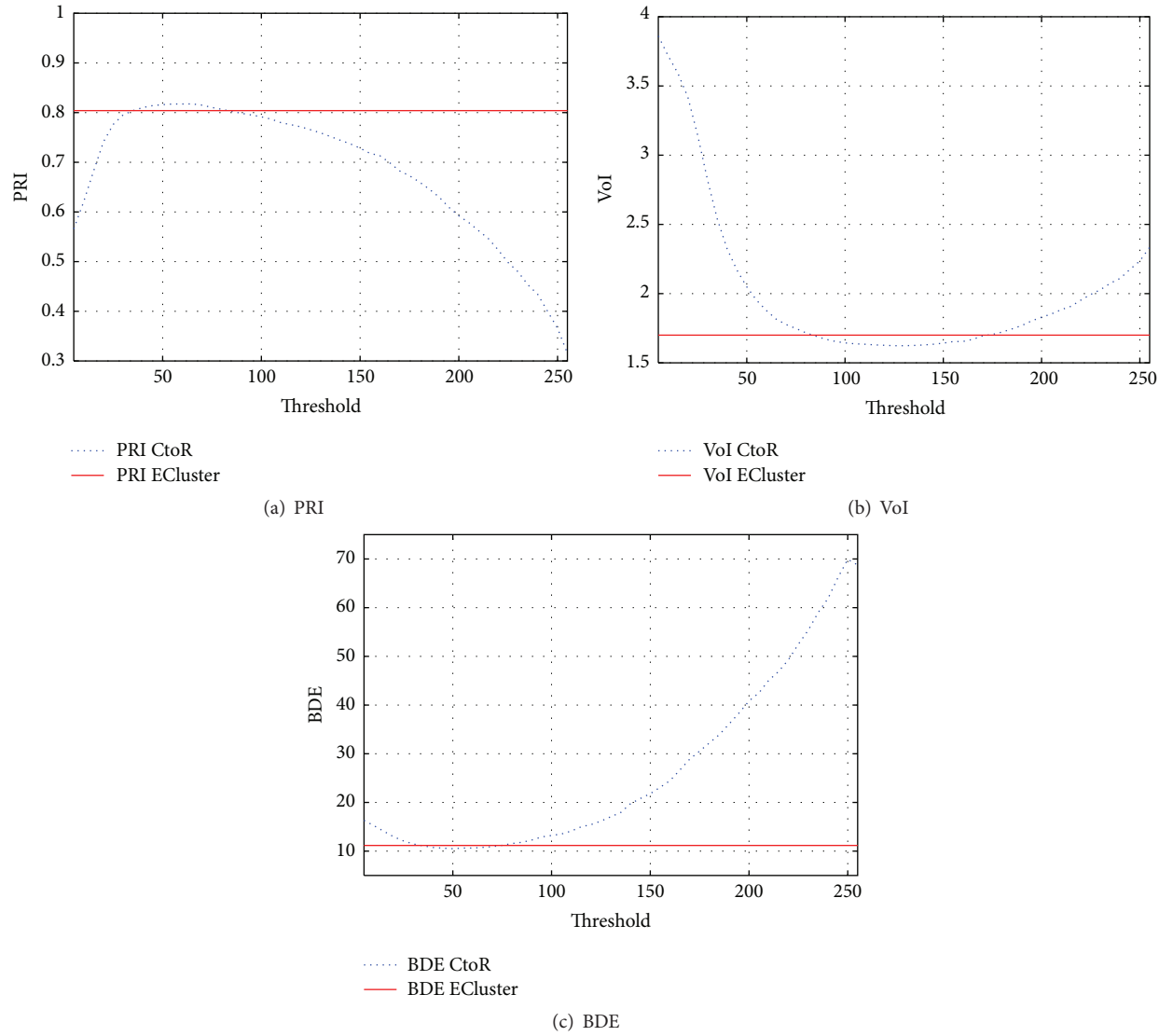


Figure 5: The segmentation evaluation (PRI, VoI, and BDE) versus threshold values is plotted for CtoR. Different threshold values give an optimal score for each segmentation evaluation approach. In contrast, the proposed EPartition is independent of the threshold values and finds the approximate optimal segmentation for all evaluation approaches.

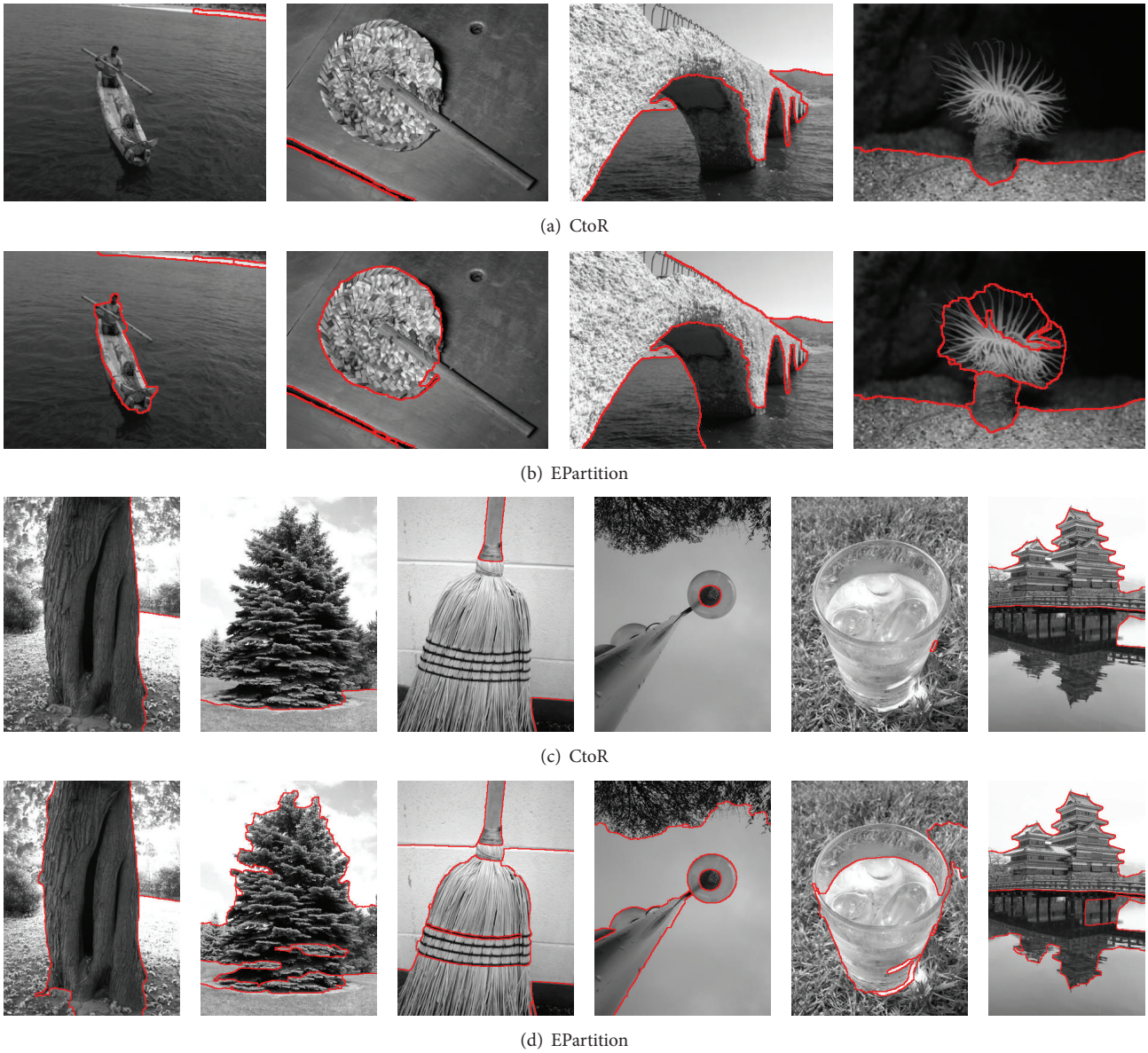
values. The proposed EPartition segmentation evaluation scores for PRI, BDE, and VoI are very close to the highest evaluation score of CtoR. However, the merge-threshold method in CtoR requires a specific threshold value for each segmentation evaluation method. The advantage of EPartition is that correct segmentation is possible without the designation of segmentation number or a threshold value.

4.2. Comparison to Trained Threshold. In previous experiments, EPartition was shown to have competitive performance with CtoR when the optimal threshold value is hand-picked for CtoR. In this section, the threshold value is trained from the Berkeley 300 set and the segmentation performances are compared to the Weizmann segmentation set [35]. The Weizmann set contains 100 images with three human segmentation annotations.

In Table 2, the segmentation evaluations of the CtoR and EPartition methods are compared. There is a minuscule difference for PRI and small differences in the VoI evaluation methods. For BDE evaluation, EPartition clearly outperforms CtoR method. The trained threshold value was not robust for different segmentation evaluation approaches. By partitioning the edges through minimizing the mean squared distance, the proposed EPartition shows adaptive performance among the three evaluation methods. Various comparative segmentation results are shown in Figure 6.

5. Conclusion and Future Works

In this paper, image segmentation by edge partitioning is proposed. In contrast with previous edge weight-based segmentation methods, such as normalized cut, the proposed



(a) CtoR

(b) EPartition

(c) CtoR

(d) EPartition

Figure 6: The segmentation results from Table 2 are shown for CtoR and the proposed EPartition.

method is independent of the number of segments. Furthermore, compared with the previous segmentation techniques, edge partitioning remains competitive without the need for the segmentation number selection. Segmentation by edge partitioning has shown to be competitive with previous segmentation techniques in the Berkeley database. The advantage of the proposed method lies in its adaptive nature for handling edge weights without threshold values or segment number assignments.

The proposed algorithm can be extended to general partitioning problems. Four labels are sufficient when segmenting 2D images. However, for fully connected graphs, the number of labels can be arbitrarily large. If a maximum number of labels are chosen, the edge partitioning method can be incorporated into a general partition problem without designating the specific number of partitions among nodes.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by Institute for Information & Communications Technology Promotion (IITP) Grant funded by the Korea government (MSIP) (no. R0101-15-0171, Development of Multimodality Imaging and 3D Simulation-Based

Table 1: Constant parameters are maintained for the Berkeley image set and the test methods. The top ranking results are written in bold, and the rankings are in parenthesis. For segmentation methods with a $*$, a different segmentation number is assigned optimally for each image.

Method	PRI	VoI	BDE
MShift [1]	0.7958	1.9725	14.41
JSEG [20]	0.7756	2.3217	14.40 (3)
GBIS [21]	0.7139	3.3949	16.67
NTP [22]	0.7521	2.4954	16.30
Saliency [23]	0.7758	1.8165	16.24
TBES [24]	0.80 (3)	1.76 (3)	—
CtoR [25]	**0.806** (1)	1.717 (2)	11.57 (2)
EPartition	0.804 (2)	**1.700** (1)	**11.16** (1)
NCut* [11]	0.772	2.259	12.94
SpecSeg* [13]	0.8146	1.8545	12.21

Table 2: The segmentation results for the Weizmann image set are summarized. The threshold value of CtoR is trained from the Berkeley set. Superior results are written in bold.

Method	PRI	VoI	BDE
CtoR [25]	**0.7170**	**1.017**	22.01
EPartition	0.7169	1.225	**13.67**

Integrative Diagnosis-Treatment Support Software System for Cardiovascular Diseases). This work was also supported by Hankuk University of Foreign Studies Research Fund.

References

[1] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[2] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV '03)*, vol. 1, pp. 10–17, IEEE, Nice, France, October 2003.

[3] A. Fabijańska and J. Gocławski, "New accelerated graph-based method of image segmentation applying minimum spanning tree," *IET Image Processing*, vol. 8, no. 4, pp. 239–251, 2014.

[4] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2281, 2012.

[5] Z. Tu, X. Chen, A. L. Yuille, and S.-C. Zhu, "Image parsing: unifying segmentation, detection, and recognition," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 113–140, 2005.

[6] J. Puzicha, T. Hofmann, and J. M. Buhmann, "Non-parametric similarity measures for unsupervised texture segmentation and image retrieval," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 267–272, IEEE, June 1997.

[7] Y. Gdalyahu, D. Weinshall, and M. Werman, "Stochastic image segmentation by typical cuts," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Fort Collins, Colo, USA, June 1999.

[8] N. Shental, A. Zomet, T. Hertz, and Y. Weiss, "Learning and inferring image segmentations using the GBP typical cut algorithm," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, vol. 2, pp. 1243–1250, Nice, France, October 2003.

[9] O. O. Olugbara, E. Adetiba, and S. A. Oyewole, "Pixel intensity clustering algorithm for multilevel image segmentation," *Mathematical Problems in Engineering*, vol. 2015, Article ID 649802, 19 pages, 2015.

[10] E. Cuevas, A. González, F. Fausto, D. Zaldívar, and M. Pérez-Cisneros, "Multithreshold segmentation by using an algorithm based on the behavior of locust swarms," *Mathematical Problems in Engineering*, vol. 2015, Article ID 805357, 25 pages, 2015.

[11] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[12] R. Zabih and V. Kolmogorov, "Spatially coherent clustering using graph cuts," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. II-437–II-444, IEEE, June-July 2004.

[13] T. H. Kim, K. M. Lee, and S. U. Lee, "Learning full pairwise affinities for spectral segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2101–2108, IEEE, San Francisco, Calif, USA, June 2010.

[14] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[15] S. Osher and N. Paragios, *Geometric Level Set Methods in Imaging, Vision, and Graphics*, Springer, New York, NY, USA, 2004.

[16] S. Chopra and M. R. Rao, "The partition problem," *Mathematical Programming*, vol. 59, no. 1, pp. 87–115, 1993.

[17] E. D. Demaine and N. Immorlica, "Correlation clustering with partial information," in *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques*, vol. 2764 of *Lecture Notes in Computer Science*, pp. 1–13, Springer, Berlin, Germany, 2003.

[18] S. Nowozin and S. Jegelka, "Solution stability in linear programming relaxations: graph partitioning and unsupervised learning," in *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*, Montreal, Canada, June 2009.

[19] E. Hodneland, X.-C. Tai, and H.-H. Gerdes, "Four-color theorem and level set methods for watershed segmentation," *International Journal of Computer Vision*, vol. 83, no. 3, pp. 264–283, 2009.

[20] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 800–810, 2001.

[21] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[22] J. Wang, Y. Jia, X.-S. Hua, C. Zhang, and L. Quan, "Normalized tree partitioning for image segmentation," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.

[23] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof, "Saliency driven totla variation segmentation," in *Proceedings of the IEEE 12th International Conference on Computer Vision*, pp. 817–824, Kyoto, Japan, October 2009.

[24] S. Rao, H. Mobahi, A. Yang, S. Sastry, and Y. Ma, "Natural image segmentation with adaptive texture and boundary encoding," in *Computer Vision—ACCV 2009: 9th Asian Conference on Computer Vision, Xi'an, September 23–27, 2009, Revised Selected Papers, Part I*, vol. 5994 of *Lecture Notes in Computer Science*, pp. 135–146, Springer, Berlin, Germany, 2010.

[25] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "From contours to regions: an empirical evaluation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 2294–2301, IEEE, Miami, Fla, USA, June 2009.

[26] K. Appel and W. Haken, "Solution of the four color map problem," *Scientific American*, vol. 237, no. 4, pp. 108–121, 1977.

[27] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[28] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing binary MRFs via extended roof duality," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, IEEE, Minneapolis, Minn, USA, June 2007.

[29] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-class Object Recognition and Segmentation," in *Computer Vision-ECCV 2006*, vol. 3951 of *Lecture Notes in Computer Science*, pp. 1–15, Springer, Berlin, Germany, 2006.

[30] M. Maire, P. Arbeláez, C. Fowlkes, and J. Malik, "Using contours to detect and localize junctions in natural images," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.

[31] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 929–944, 2007.

[32] M. Meila, "Comparing clustering: an axiomatic view," in *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, pp. 577–584, ACM Press, 2005.

[33] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí, "Yet another survey on image segmentation: region and boundary information integration," in *Computer Vision ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part III*, vol. 2352 of *Lecture Notes in Computer Science*, pp. 408–422, Springer, Berlin, Germany, 2002.

[34] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of the 8th International Conference on Computer Vision*, pp. 416–423, July 2001.

[35] S. Alpert, M. Galun, R. Basri, and A. Brandt, "Image segmentation by probabilistic bottom-up aggregation and cue integration," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, Minneapolis, Minn, USA, June 2007.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 629254, 7 pages
http://dx.doi.org/10.1155/2015/629254

Research Article

Clustering of Rainfall Stations in RH-24 Mexico Region Using the Hurst Exponent in Semivariograms

Francisco Gerardo Benavides-Bravo,[1] F-Javier Almaguer,[2] Roberto Soto-Villalobos,[2] Víctor Tercero-Gómez,[3] and Javier Morales-Castillo[2]

[1]*Instituto Tecnológico de Nuevo León, 67170 Guadalupe, NL, Mexico*
[2]*Universidad Autónoma de Nuevo León, 66451 San Nicolás de los Garza, NL, Mexico*
[3]*Tecnológico de Monterrey, 64849 Monterrey, NL, Mexico*

Correspondence should be addressed to Francisco Gerardo Benavides-Bravo; fgbenavid@gmail.com

Received 15 August 2015; Revised 14 November 2015; Accepted 23 November 2015

Academic Editor: Costas Panagiotakis

An important topic in the study of the time series behavior and, in particular, meteorological time series is the long-range dependence. This paper explores the behavior of rainfall variations in different periods, using long-range correlations analysis. Semivariograms and Hurst exponent were applied to historical data in different pluviometric stations of the Río Bravo-San Juan watershed, at the hydrographic RH-24 Mexico region. The database was provided by the Water National Commission (CONAGUA). Using the semivariograms, the Hurst exponent was obtained and used as an input to perform a cluster analysis of rainfall stations. Groups of homogeneous samples that might be useful in a regional frequency analysis were obtained through the process.

1. Introduction

When limited observations of hydrological events are available, the ability to provide appropriate characterization, analysis, and predictions of a phenomenon gets compromised. However, the analysis can be improved by identifying homogeneous samples that can be used in combination to make better estimates of a probability model. This is one of the major concerns within the practice of regional frequency analysis (RFA), where the final output is the estimation of extreme events in a geographical area that can be used as input in risk analysis, water management, zoning, and land use applications, Hosking and Wallis [1]. However, the estimation of extreme events is considered a complex problem, mostly because the information is usually limited, serial correlation exists, multiple change-points might be present, and observations follow trends and seasonal patterns. To address these issues, hydrological time series studies have been successfully applied in the past, Machiwal and Jha [2]; however, most research efforts have been focused on trend detection tests, leaving aside other important properties such as stationarity, homogeneity, periodicity, and persistence. By addressing these properties, a better selection of homogeneous samples might be possible, and as a consequence, practitioners might achieve better predictions.

Previous works on time series analysis in climatology with applications in precipitation go back to Bhuiya [3], with the development of a test for stationarity after periodic and trend components were subtracted from hydrologic series. Buishand [4] used trend tests to evaluate the difference in precipitation between rural and urban areas of Amsterdan and Rotterdam. Buishand [5, 6] constructed several tests of homogeneity in the mean of series with the use of cumulative sums, likelihood tests, and Bayesian inference. Kothyari et al. [7] evaluated three stations in India, Agra, Dehradun, and Dehli, to test for changes in rainfall and temperature, providing evidence of a change in the number of rainy days during monsoon season and an increment in temperature. Giakoumakis and Baloutsos [8] performed a trend analysis on historical series of annual precipitations from the basin of the Evinos Riven in Greece. By applying different tests of randomness, decreasing trends were found in the rainfall

records. Other authors dealing with trend analysis, homogeneity, and change-points found in the literature are Angel and Huff [9], Mirza et al. [10], Tarhule and Woo [11], Luís et al. [12], Kripalani and Kulkarni [13], Adamowski and Bougadis [14], Yu et al. [15], and Kumar et al. [16]. A comprehensive review of these works can be found in Machiwal and Jha [2] with descriptions of related developments in hydrological time series analysis.

Recent developments in hydrological analysis include the works of Golian et al. [17] with a classification and clustering approach of rainfall data using the natural-breaks classification method and the fuzzy c-means (FCM) algorithm. Shi et al. [18] analyzed variations in trends for precipitation data using a linear regression method, the Mann-Kendall test, and the Hurst exponent. The Hurst exponent, as part of a fractal analysis, was used to evaluate long-range dependence and the possibility of trends in the data. The following works around the Hurst exponent include the developments of Golder et al. [19], where the Hurst exponent is also used to explore long-term correlations, and cumulative rainfall observations were modeled using the alpha-stable probability law to deal with heavy-tailed distributions. Chang [20] extended the application of the Hurst exponent by developing a computation approach to estimate the exponent over time series that fits a discrete time fractional Brownian motion and fractional Gaussian noise. Yu et al. [21] also studied long-term correlations using the Hurst exponent and performed a multifractal analysis of rainfall series (see Kantelhardt [22]) based on a multiplicative cascade model and a multifractal detrended fluctuation analysis. Other recent works on time series analysis can be found in Carbone et al. [23] with the construction of a simulation model of storms using a double exponential distribution. Chou [24] investigated the complexity at different temporal scales of rainfall and runoff time series using the sample-entropy method, and finally, García-Marín et al. [25] performed a regional frequency analysis over rainfall data from Málaga, Spain, where the grouping of stations into homogenous regions has been done by following a cluster analysis with multifractal values of the different series.

In this paper, a clustering approach is used to group stations into homogeneous samples after summarizing the results of semivariogram analysis into a Hurst exponent. As a case study, a sample of pluviometric stations, the Río Bravo-San Juan watershed, at the RH-24 Mexico region was analyzed (Figure 1). A map of the Río Bravo-San Juan watershed is shown in Figure 2. This region is located in Mexico between the states of Nuevo León, Coahuila, and Tamaulipas covering an approximate area of 29420 km^2. Some of the rainfall stations are shown in Figure 3. The data used has been provided by CONAGUA, the local institution responsible for water management in the country.

2. Problem Description

In practice, to perform RFA, it is required to identify homogeneous regions where data follow similar patterns that can be analyzed together to improve the identification of probability models that in turn can be used to estimate extreme events and their frequency in terms of return periods. This analysis is usually executed when dealing with droughts, pollution, wind movement, temperature, atmospheric pressure, and rainfall observations, to name a few. These researches deal with the problem of finding groups of rainfall stations that create homogeneous regions by considering fractal structures captured through semivariograms and Hurst exponents. Rainfall data from a sample of the hydrographic region RH-24 Mexico, the Río Bravo-San Juan watershed, are used as a case study to evaluate the proposed approach.

Figure 1: The watersheds of Mexico with Google Earth. In black edge, the Río Bravo-San Juan watershed. Source of the database: http://www.conagua.gob.mx/.

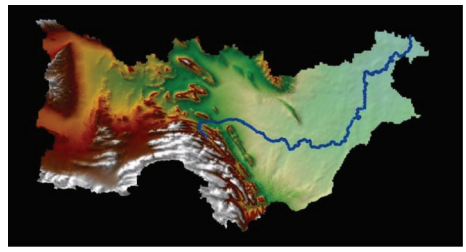


Figure 2: Río Bravo-San Juan watershed (San Juan river in blue). Image source: "Water Management in the Río San Juan Watershed, in the Southern Río Bravo Hydrologic Region of Mexico" at http://earthzine.org/2012/08/13/.

3. Methodology

Semivariograms, in the present study, are used to quantify long-range correlations of data from different pluviometric stations using monthly records. By considering the analysis of semivariograms of historical series, a rescaled range analysis R/S is performed to obtain a measure of the Hurst exponent [26]. The Hurst exponent is used as a metric of a particular pluviometric station. The process is repeated over each pluviometric station within the region under analysis. Hurst exponents are used as a reference to identify stations that exhibit similar patterns. As a consequence, a cluster analysis is applied to identify homogeneous samples. An advantage of the Hurst exponent is the simplicity of its algorithm that can be used to measure the condition of persistence or antipersistence of a process, and it provides a metric that can be used to classify different time series.

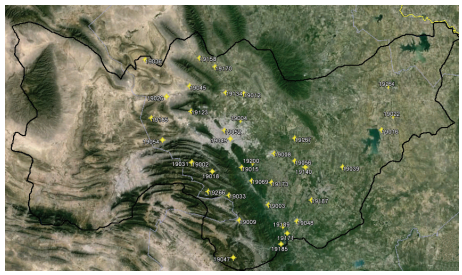


Figure 3: Geographical locations (from Google Earth) of the pluviometric stations at the Río Bravo-San Juan watershed. Source of the database: http://www.conagua.gob.mx/.

3.1. Semivariogram. The semivariogram or variogram $\gamma(h)$ is used to describe the relationship of paired observations separated by a distance h. It is a geostatistical technique that allows a quantitative measure of the long-range persistence in nonstationary time series Witt and Malamud [27], Haslett [28], and Dmowska and Saltzman [29]. Correlations over time and space create patterns that can be used to describe the behavior of a set of observations. Mathematically, the variogram estimates the expected squared difference between neighboring random variables. This calculation is performed over different h values. Given a time series or stochastic processes $\{X_t,\ t \geq 0\}$, the autocovariance function at the point $(t, t+h)$ is defined as $C_X(t, t+h) = E[X_t X_{t+h}] - E[X_t]E[X_{t+h}]$, with $E[X_t]$ as the mean of the process at time t. The semivariogram $\gamma(h)$ is given by half of the variance of the difference between pairs of observations at different "locations" in time:

$$\begin{aligned}\gamma(h) &= \frac{1}{2}\mathrm{Var}\left(X_{t+h} - X_t\right)\\ &= \frac{1}{2}E\left[\left(X_{t+h} - X_t - E\left[X_{t+h} - X_t\right]\right)^2\right]\\ &= \frac{\mathrm{Var}\left(X_t\right) + \mathrm{Var}\left(X_{t+h}\right)}{2} - \mathrm{Cov}\left(X_t, X_{t+h}\right)\\ &= \frac{\mathrm{Var}\left(X_t\right) + \mathrm{Var}\left(X_{t+h}\right)}{2}\\ &\quad - \sqrt{\mathrm{Var}\left(X_t\right)\mathrm{Var}\left(X_{t+h}\right)}\rho_X(t, t+h),\end{aligned} \tag{1}$$

where $-1 \leq \rho_X(t, t+h) \leq 1$ is the autocorrelation function (the autocovariance function normalized).

In the special case when $\rho(x_{t+h}, x_t) = 0$, $\forall(t, h)$, it is said that the stochastic processes $\{X_t,\ t \geq 0\}$ are uncorrelated and the semivariogram is reduced to the arithmetic mean of the variance of processes at times t and $t+h$:

$$\gamma(h) = \frac{\mathrm{Var}\left(X_t\right) + \mathrm{Var}\left(X_{t+h}\right)}{2}. \tag{2}$$

If the random field $\{X_t, t \geq 0\}$ has constant mean $= E[X_t] = E[X_{t+h}] = \mu\ \forall t$, the semivariogram (1) adopts the simple form:

$$\gamma(h) = \frac{1}{2}E\left[\left(X_{t+h} - X_t\right)^2\right]. \tag{3}$$

If $X(t)$ and $X(t+h)$ are independent random variables $\forall t$, again, though for a different reason, the semivariogram is reduced to the special case (2).

In principle, given a stochastic process $\{X_t,\ t \geq 0\}$, the expected value of differences $E[X_{t+h} - X_t]$ at time t and with lag h is empirically estimated by the average over a "large enough" ensemble of realizations or paths in time. However, for a single time series $\{X_n,\ n = 1, 2, \ldots, n\}$, the expected value can be estimated assuming an ergodic hypothesis, that is, a statistical principle of equivalence according to which *the average over time and the average over the ensemble are the same*, Lefebvre [30]. Thereby the differences $X_{t+h} - X_t$, which would be obtained with an infinitely reproducible process, are "simulated" or "cloned" from the "mother series." Thus, the average value of differences $X_{t+h} - X_t$ is estimated by

$$E\left[X_{t+h} - X_t\right] = \frac{1}{n(h)}\sum_{i=1}^{n(h)}\left(x_{i+h} - x_i\right), \tag{4}$$

where $n(h)$ is the number of differences with a lag h. When $h = 1, 2, 3, \ldots$, the averages (4) are, respectively,

$$\begin{aligned}E\left[X_{t+1} - X_t\right] &= \frac{x_n - x_1}{n-1},\\ E\left[X_{t+2} - X_t\right] &= \frac{x_{n-1} + x_n - \left(x_1 + x_2\right)}{n-2},\\ E\left[X_{t+3} - X_t\right] &= \frac{x_{n-2} + x_{n-1} + x_n - \left(x_1 + x_2 + x_3\right)}{n-3},\\ &\vdots\\ E\left[X_{t+h} - X_t\right] &= \frac{1}{n(1-h/n)}\left(\sum_{j=0}^{k-1}x_{n-j} - \sum_{l=1}^{k}x_l\right).\end{aligned} \tag{5}$$

According to (5) for a maximum value of h "relatively moderate" or $h/n < 1$, except in the presence of isolated extreme outliers, the two summations in (5) are roughly of the same order, such that the empirical average value (4) can be approximated by $E[X_{t+h}] \approx E[X_t] = m =$ constant. This is an observed characteristic in the time series of the pluviometric stations. Therefore, the corresponding estimator of (3) is simply

$$\gamma(h) = \frac{1}{2n(h)}\sum_{t=1}^{n(h)}\left(x_{t+h} - x_t\right)^2. \tag{6}$$

3.2. Measurement of the Hurst Exponent H. To estimate the Hurst exponent from a temporal series $\{X_k\}$, with $k \in 1, 2, \ldots, N$, the series is divided in a group of d-subseries of length m. Really, the size m is an average number. A standard way, though not the only, to obtain the m size of the subseries is partitioning the original series in powers of base 2. In doing so, in each of the successive partitions, the approximate value

of m is as follows: $N, N/2, N/2^2, N/2^3, \ldots$. For each subseries $n = 1, 2, \ldots, d$, do the following:

(1) Calculate the mean E_n and the standard deviation S_n.

(2) Calculate the deviation with respect to the mean by subtracting the mean of each element using

$$Z_{in} = X_{in} - E_n, \quad i = 1, 2, \ldots, m. \tag{7}$$

(3) Get the partial sums:

$$Y_{in} = \sum_{j=1}^{i} Z_{jn}, \quad i = 1, 2, \ldots, m. \tag{8}$$

(4) Calculate the range:

$$R_n = \max_{i=1:m}(Y_{in}) - \min_{i=1:m}(Y_{in}). \tag{9}$$

(5) Normalize the range:

$$\frac{R_n}{S_n}. \tag{10}$$

(6) For each subseries of length m take the average:

$$\left\langle \frac{R}{S} \right\rangle_m = \frac{1}{d}\sum_{n=1}^{d} \frac{R_n}{S_n}. \tag{11}$$

(7) Hurst [26] found the relation of the statistical $\langle R/S \rangle_m$ given by the following power law:

$$\left\langle \frac{R}{S} \right\rangle_m \approx cm^H, \tag{12}$$

where H is the Hurst exponent and c is a positive constant.

Two factors involved in the determination of the Hurst coefficient are the way time series is divided into a group of subseries and the asymptotic behavior of the rescaled range. First, the range of values m are used to calculate the slope of $\log(\langle R/S \rangle_m)$ given the relationship

$$\log\left(\left\langle \frac{R}{S} \right\rangle_m\right) = \log(c) + H\log(m). \tag{13}$$

Second, the determination of H is the result of the asymptotic behavior of the rescaled range, that is, when the value m tends to infinity. The analysis of the rescaled $\langle R/S \rangle_m$ over some values of m is estimated using a log/log expression given in (13). To obtain H coefficient, the least-squares method is used. The slope of this line is the Hurst coefficient H.

This exponent is considered a fractal index, Mandelbrot and Wallis [31], and provides information about long-term correlations exhibited by a series of observations; for a theoretical review of the Hurst exponent, see Mandelbrot [32]. In practice, the Hurst exponent can take values between 0 and 1, where

(i) $0 < H < 0.5$ indicates nonpersistency in a series; that is, an increment is more likely to be followed by a decrement and vice versa;

(ii) $H = 0.5$ indicates lack of serial correlation (Gaussian white noise);

(iii) $0.5 < H < 1$ indicates persistency; that is, an increment is likely to be followed by an increment and a decrement by another decrement.

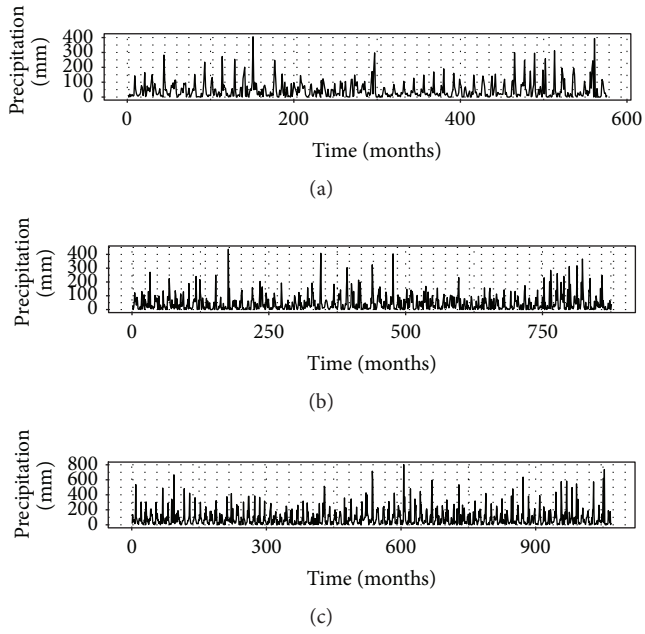


Figure 4: Time series for rainfall measurements from three stations at Río Bravo-San Juan watershed, Mexico. From the top to the bottom, respectively: Apodaca, station number 19004, 1940 January–2012 December; El Cuchillo, station number 19016, 1939 January–2012 December; La Boca, station number 19069, 1923 January–2012 December.

4. Results

To illustrate the procedure, only the analysis of three rainfall stations was selected to be presented in this section. The measured values of monthly precipitation in millimeters from the stations Apodaca, El Cuchillo, and La Boca are displayed in Figure 4, and results obtained taking into account all stations (following the same process) are presented at the end of this section.

As can be seen, patterns and relationships between different stations are difficult to assess only through "eyeball" analysis. However, when semivariograms are obtained, as shown in Figure 5, a footprint of the data becomes more evident. A closer inspection in every station shows a seasonal pattern that repeats every 12 observations in the semivariogram. This can be explained due to the fact that monthly observations were used in the analysis.

Once the semivariograms were obtained, the Hurst exponent for the series of the γ values was calculated. It can be seen that Hurst coefficients are close to 1, which indicates a positive long dependency of the data in the semivariograms. Hurst exponents from all stations under analysis are presented in Table 1, where the long dependency in all variograms becomes clear. Some of the coefficients that appear in the table exceed the interval established of possible values of the Hurst exponent, $0 < H < 1$; this is a known error due to estimation bias or a possible linear retrogression.

To address the issue of finding homogeneous samples, a cluster analysis is performed using estimates of the Hurst exponent. As shown in Figure 6, a histogram of frequencies

Table 1: Hurst coefficients for semivariogram (6) from pluviometric stations at Río Bravo-San Juan watershed.

Station	Name	Latitude	Longitude	Data	Hurst
19015	El Cerrito	25 30 36	100 11 36	1939–2012	0.71245
19039	Las Enramadas	25 30 05	099 31 17	1940–2012	0.78917
19018	El Pajonal	25 29 23	100 23 20	1955–2012	0.79764
19012	Ciénega de Flores	25 57 08	100 10 20	1940–2012	0.80106
19003	Allende	25 17 01	100 01 13	1940–2012	0.82492
19069	La Boca	25 25 46	100 07 44	1923–2013	0.8387
19189	El Pastor	25 09 06	099 55 36	1987–2011	0.84065
19009	Casillas	25 11 47	100 12 51	1956–2012	0.84258
19187	California	25 18 23	099 44 02	1982–2011	0.84406
19173	Palmitos	25 25 02	099 59 50	1982–2012	0.85304
19002	Agua Blanca	25 32 39	100 31 23	1958–2012	0.86171
19134	Salinas Victoria	25 57 33	100 17 34	1979–2011	0.86501
19031	La Cruz	25 32 47	100 31 23	1955–2011	0.86751
19036	La Popa	26 09 50	100 49 40	1956–2011	0.87324
19008	Cadereyta Jiménez	25 35 25	099 58 30	1995–2012	0.87335
19185	El Canada	25 02 48	099 56 29	1982–2011	0.87351
19056	San Juan	25 32 36	099 50 25	1944–2012	0.87678
19016	El Cuchillo	25 43 05	099 15 21	1960–2012	0.87778
19052	Monterrey(Obs)	25 44 01	100 16 01	1986–2008	0.87802
19026	Icamole	25 56 28	100 41 13	1954–2012	0.88481
19200	La Cienega	25 32 10	100 07 15	1984–2011	0.89216
19004	Apodaca	25 47 37	100 11 50	1964–2012	0.89506
19047	Mimbres	24 58 26	100 15 31	1957–2011	0.89837
19140	Tepehuaje	25 30 19	099 46 15	1979–2012	0.91092
19267	Santa Ma. La Floreña	25 10 59	99 46 00	1984–2011	0.91686
19048	Montemorelos	25 10 55	099 49 56	1940–2012	0.91836
19040	Los Aldama	26 03 52	099 11 48	1942–1994	0.92157
19022	General Bravo	25 48 05	099 10 32	1927–2012	0.92301
19158	Rancho de Gomas	26 10 11	100 27 52	1981–2011	0.93016
19045	Mina	26 00 08	100 32 00	1953–2012	0.93308
19054	Rinconada	25 40 52	100 43 03	1945–2012	0.93977
19165	Chupaderos del Indio	25 48 49	100 47 24	1982–2011	0.94288
19266	San Jose de Barranquillas	26 32 41	100 28 21	1978–2011	0.94911
19171	Lampacitos	25 06 38	099 53 57	1982–2011	0.95188
19264	Dr. Coss	25 51 16	099 56 36	1982–2011	0.96461
19170	El Hojase	26 06 55	100 21 38	1982–2011	1.00650

was used to separate stations into 6 clusters. These clusters and the result of fitting probability distributions over the data of each rainfall station are shown in Table 2.

Distributions were selected based on a goodness of fit analysis. After identifying a set of feasible distributions with p values bigger than a significant level of 0.05, in every case, the distribution with the highest average p value was selected for each station. Gamma and Generalized Extreme Value were the distributions that gave the best fit over the data analyzed. These functions are

$$f(x;\alpha,\beta) = \frac{x^{\alpha-1}}{\beta^{\alpha}\Gamma(\alpha)} e^{-x/\beta}, \tag{14}$$

$$f(x;k,\sigma,\mu) = \frac{1}{\sigma} e^{-(1+kz)^{-1/k}} (1+kz)^{-1-1/k}, \qquad z = \frac{x-\mu}{\sigma}, \tag{15}$$

respectively.

As a result of the analysis, estimated distributions do not mix within cluster. This input can be used in a posterior analysis to obtain maximum likelihood estimators of the distribution parameters using all data concurrently.

5. Conclusions

Variograms followed by analysis of R/S or Hurst exponent estimation were used as input of a cluster analysis. Hurst

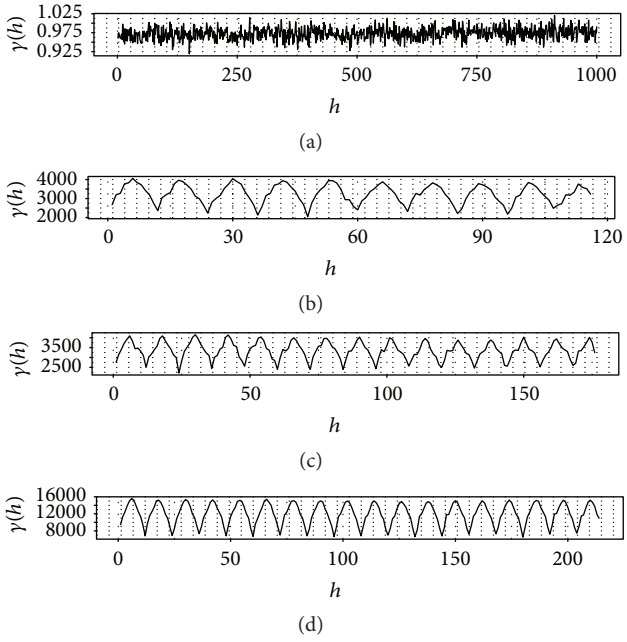


Figure 5: Semivariogram (6) corresponding to the time series of rainfall measurements from the three stations of the Río Bravo-San Juan watershed shown in Figure 4. In each case the maximum lag $h_{\max} = 0.20 * N$ was used, with N indicating the size of the time series. For comparison, the semivariogram corresponding to Gaussian white noise is included. From top to bottom, respectively: Gaussian white noise, Apodaca (station number 19004), El Cuchillo (station number 19016), and La Boca (station number 19069).

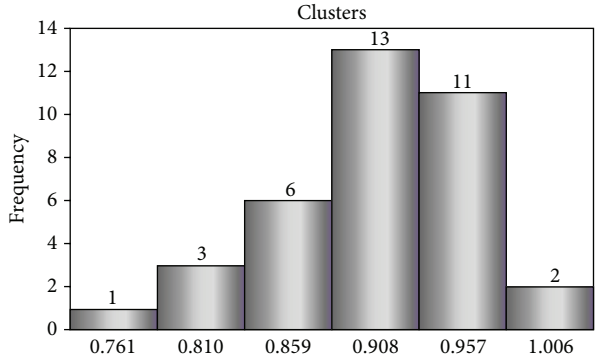


Figure 6: Histogram for the Hurst exponents of the analyzed pluviometric stations.

Table 2: Clustering of pluviometric stations by the Hurst exponent of the semivariogram.

Cluster	Station	Hurst	Distribution	Parameters
1	19015	0.71245	Gamma(α, β)	(0.7022, 121.06)
3	19039	0.78917	Gamma(α, β)	(0.8822, 101.75)
	19018	0.79764		(0.5731, 78.07)
	19012	0.80106		(0.5402, 119.7)
6	19003	0.82492	GEV(k, σ, μ)	(0.349, 43.609, 38.143)
	19069	0.83870		(0.389, 43.321, 33.483)
	19189	0.84065		(0.453, 35.036, 22.915)
	19009	0.84258		(0.376, 25.309, 17.281)
	19187	0.84406		(0.377, 31.132, 24.640)
	19173	0.85304		(0.379, 32.826, 25.056)
13	19002	0.86171	GEV(k, σ, μ)	(0.351, 27.643, 20.471)
	19134	0.86501		(0.319, 21.703, 16.845)
	19031	0.86751		(0.313, 33.237, 23.081)
	19036	0.87324		(0.506, 9.5213, 5.0588)
	19008	0.87335		(0.407, 28.933, 22.421)
	19185	0.87351		(0.379, 22.415, 16.03)
	19056	0.87678		(0.372, 30.176, 22.56)
	19016	0.87778		(0.366, 24.295, 16.683)
	19052	0.87802		(0.429, 25.864, 19.559)
	19026	0.88481		(0.418, 8.7536, 5.424)
	19200	0.89216		(0.407, 33.194, 23.434)
	19004	0.89506		(0.368, 23.938, 17.563)
	19047	0.89837		(0.168, 33.305, 28.427)
11	19140	0.91092	GEV(k, σ, μ)	(0.369, 30.367, 23.162)
	19267	0.91686		(0.407, 20.642, 15.472)
	19048	0.91836		(0.356, 37.599, 30.006)
	19040	0.92157		(0.356, 20.614, 13.77)
	19022	0.92301		(0.383, 25.054, 16.587)
	19158	0.93016		(0.421, 16.537, 10.853)
	19045	0.93308		(0.457, 12.116, 7.9187)
	19054	0.93977		(0.485, 9.1504, 5.2866)
	19165	0.94288		(0.434, 10.814, 6.4381)
	19266	0.94911		(0.369, 32.361, 22.162)
	19171	0.95188		(0.435, 31.356, 22.367)
2	19264	0.96461	GEV(k, σ, μ)	(0.401, 20.168, 1.2395)
	19170	1.00650		(0.480, 20.046, 12.109)

exponent provides a measure to determine if a time series is like a Gaussian white noise or has underlying trends, and it can be used to cluster the pluviometric stations according to the values of their semivariograms. As a case study to evaluate this approach, a sample of rainfall stations, those included in the Río Bravo-San Juan watershed from the hydrographic region RH-24 Mexico, was used in the analysis. Long-range dependency was found in every variogram evaluated with the Hurst exponent; however, it was still found useful as an input of a cluster analysis. A goodness of fit process was executed with every series, and the results showed that existing dominant distributions within a feasible set (found independently in each station) do not overlap over clusters. The probability distributions were found nested within each cluster. This is indicative that homogeneous patterns were identified within groups, and groups were heterogeneous between themselves.

The study of the rainfall stations with semivariograms and R/S analysis provides a powerful tool that allows practitioners to analyze long-term correlations and clustering in hydrological time series. In future work, L-moments and spectral and wavelets analysis will be used to improve understanding of complex time series of pluviometric rainfall levels.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] J. R. M. Hosking and J. R. Wallis, *Regional Frequency Analysis: An Approach Based on L-Moments*, 2005.

[2] D. Machiwal and M. K. Jha, "Current status of time series analysis in hydrological sciences," in *Hydrologic Time Series Analysis: Theory and Practice*, pp. 96–136, Springer Netherlands, 2012.

[3] R. K. Bhuiya, "Stochastic analysis of periodic hydrologic process," *Journal of the Hydraulics Division*, vol. 97, no. 7, pp. 949–962, 1971.

[4] T. A. Buishand, "Urbanization and changes in precipitation, a statistical approach," *Journal of Hydrology*, vol. 40, no. 3-4, pp. 365–375, 1979.

[5] T. A. Buishand, "Some methods for testing the homogeneity of rainfall records," *Journal of Hydrology*, vol. 58, no. 1-2, pp. 11–27, 1982.

[6] T. A. Buishand, "Tests for detecting a shift in the mean of hydrological time series," *Journal of Hydrology*, vol. 73, no. 1-2, pp. 51–69, 1984.

[7] U. C. Kothyari, V. P. Singh, and V. Aravamuthan, "An investigation of changes in rainfall and temperature regimes of the Ganga Basin in India," *Water Resources Management*, vol. 11, no. 1, pp. 17–34, 1997.

[8] S. G. Giakoumakis and G. Baloutsos, "Investigation of trend in hydrological time series of the Evinos River basin," *Hydrological Sciences Journal*, vol. 42, no. 1, pp. 81–88, 1997.

[9] J. R. Angel and F. A. Huff, "Changes in heavy rainfall in Midwestern United States," *Journal of Water Resources Planning and Management*, vol. 123, no. 4, pp. 246–249, 1997.

[10] M. Q. Mirza, R. A. Warrick, N. J. Ericksen, and G. J. Kenny, "Trends and persistence in precipitation in the Ganges, Brahmaputra and Meghna river basins," *Hydrological Sciences Journal*, vol. 43, no. 6, pp. 845–858, 1998.

[11] A. Tarhule and M.-K. Woo, "Changes in rainfall characteristics in northern Nigeria," *International Journal of Climatology*, vol. 18, no. 11, pp. 1261–1271, 1998.

[12] M. D. Luís, J. Raventós, J. C. González-Hidalgo, J. R. Sánchez, and J. Cortina, "Spatial analysis of rainfall trends in the region of Valencia (East Spain)," *International Journal of Climatology*, vol. 20, no. 12, pp. 1451–1469, 2000.

[13] R. H. Kripalani and A. Kulkarni, "Monsoon rainfall variations and teleconnections over South and East Asia," *International Journal of Climatology*, vol. 21, no. 5, pp. 603–616, 2001.

[14] K. Adamowski and J. Bougadis, "Detection of trends in annual extreme rainfall," *Hydrological Processes*, vol. 17, no. 18, pp. 3547–3560, 2003.

[15] P.-S. Yu, T.-C. Yang, and C.-C. Kuo, "Evaluating long-term trends in annual and seasonal precipitation in Taiwan," *Water Resources Management*, vol. 20, no. 6, pp. 1007–1023, 2006.

[16] V. Kumar, S. K. Jain, and Y. Singh, "Analysis of long-term rainfall trends in India," *Hydrological Sciences Journal*, vol. 55, no. 4, pp. 484–496, 2010.

[17] S. Golian, B. Saghafian, S. Sheshangosht, and H. Ghalkhani, "Comparison of classification and clustering methods in spatial rainfall pattern recognition at Northern Iran," *Theoretical and Applied Climatology*, vol. 102, no. 3, pp. 319–329, 2010.

[18] P. Shi, X. Ma, X. Chen, S. Qu, and Z. Zhang, "Analysis of variation trends in precipitation in an upstream catchment of Huai River," *Mathematical Problems in Engineering*, vol. 2013, Article ID 929383, 11 pages, 2013.

[19] J. Golder, M. Joelson, M.-C. Neel, and L. Di Pietro, "A time fractional model to represent rainfall process," *Water Science and Engineering*, vol. 7, no. 1, pp. 32–40, 2014.

[20] Y.-C. Chang, "Efficiently implementing the maximum likelihood estimator for Hurst exponent," *Mathematical Problems in Engineering*, vol. 2014, Article ID 490568, 10 pages, 2014.

[21] Z.-G. Yu, Y. Leung, Y. D. Chen, Q. Zhang, V. Anh, and Y. Zhou, "Multifractal analyses of daily rainfall time series in Pearl River basin of China," *Physica A: Statistical Mechanics and Its Applications*, vol. 405, pp. 193–202, 2014.

[22] J. W. Kantelhardt, "Fractal and multifractal time series," in *Encyclopedia of Complexity and Systems Science*, pp. 3754–3779, Springer, New York, NY, USA, 2009.

[23] M. Carbone, M. Turco, G. Brunetti, and P. Piro, "A cumulative rainfall function for subhourly design storm in Mediterranean Urban areas," *Advances in Meteorology*, vol. 2015, Article ID 528564, 10 pages, 2015.

[24] C.-M. Chou, "Complexity analysis of rainfall and runoff time series based on sample entropy in different temporal scales," *Stochastic Environmental Research and Risk Assessment*, vol. 28, no. 6, pp. 1401–1408, 2014.

[25] A. P. García-Marín, J. Estévez, M. T. Medina-Cobo, and J. L. Ayuso-Muñoz, "Delimiting homogeneous regions using the multifractal properties of validated rainfall data series," *Journal of Hydrology*, vol. 529, pp. 106–119, 2015.

[26] H. E. Hurst, "Long-term storage capacity of reservoirs," *Transactions of the American Society of Civil Engineers*, vol. 116, no. 1, pp. 770–799, 1951.

[27] A. Witt and B. D. Malamud, "Quantification of long-range persistence in geophysical time series: conventional and benchmark-based improvement techniques," *Surveys in Geophysics*, vol. 34, no. 5, pp. 541–651, 2013.

[28] J. Haslett, "On the sample variogram and the sample autocovariance for non-stationary time series," *Journal of the Royal Statistical Society—Series D: The Statistician*, vol. 46, no. 4, pp. 475–485, 1997.

[29] R. Dmowska and B. Saltzman, Eds., *Advances in Geophysics*, Long-Range Persistence in Geophysical Time Serie, Academic Press, 1999.

[30] M. Lefebvre, *Applied Stochastic Processes*, Springer, New York, NY, USA, 2007.

[31] B. B. Mandelbrot and J. R. Wallis, "Robustness of the rescaled range *R*/*S* in the measurement of noncyclic long run statistical dependence," *Water Resources Research*, vol. 5, no. 5, pp. 967–988, 1969.

[32] B. Mandelbrot, "Statistical methodology for nonperiodic cycles: from the covariance to *R/S* analysis," *Annals of Economic and Social Measurement*, vol. 1, no. 3, pp. 259–290, 1972.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 592549, 10 pages
http://dx.doi.org/10.1155/2015/592549

Research Article

A Contribution to the Study of Ensemble of Self-Organizing Maps

Leandro Antonio Pasa,[1] José Alfredo Ferreira Costa,[2] and Marcial Guerra de Medeiros[2]

[1]*Department of Electrical Engineering, Federal University of Technology-Paraná, 85884-000 Medianeira, PR, Brazil*
[2]*Department of Electrical Engineering, Federal University of Rio Grande do Norte, 59078-970 Natal, RN, Brazil*

Correspondence should be addressed to Leandro Antonio Pasa; pasa@utfpr.edu.br

Received 20 September 2015; Accepted 25 November 2015

Academic Editor: Costas Panagiotakis

This study presents a factorial experiment to investigate the ensemble of Kohonen Self-Organizing Maps. Clusters Validity Indexes and the Mean Square Quantization Error were used as a criterion for fusing Kohonen Maps, through three different equations and four approaches. Computational simulations were performed with traditional dataset, including those with high dimensionality, not linearly separable classes, Gaussian mixtures, almost touching clusters, and unbalanced classes, from the UCI Machine Learning Repository and from Fundamental Clustering Problems Suite, with variations in map size, number of ensemble components, and the percentage of dataset bagging. The proposed method achieves a better classification than a single Kohonen Map and we applied the Wilcoxon Signed Rank Test to evidence its effectiveness.

1. Introduction

An ensemble is a collection of individual classifiers with different parameters which may lead to a higher generalization than when it is working separately, as well as a decrease in variance model and higher noise tolerance when compared to a single component [1]. Each classifier operates independently and generates a solution that is combined by the ensemble producing a single output, as illustrated in Figure 1.

For a successful outcome of the ensemble, it is necessary that the components generalize differently and the errors introduced by each component must be uncorrelated because there is no point combining models that adopt the same procedures and assumptions [2]. Furthermore, with a greater diversity among the components, the ensemble's performance is increased [3].

For the Kohonen Self-Organizing Maps (SOM) [4], uncorrelated errors can be obtained, for example, using different sets of training or using variations of training parameters for each neural network. The combination of the outputs of the SOM maps into a single output can be made by the fusion of the neurons of the maps that compose the machine committee. These neurons must represent the same input space of the region; namely, the weight vectors of neurons to be fused should be very close.

The interest on ensemble methods has grown rapidly and we can find its applications in a variety of knowledge areas. In addition to ensemble or machine committee, there are several denominations in the literature, such as classifiers committees, combined classifiers, classifier ensemble, and multiple classifier systems.

There are many ensemble applications to solve problems in different areas. In [5] they compared merged maps with the traditional SOM for document organization and retrieval. As a criterion for combining maps, the Euclidean distance between neurons was used in order to select the neurons that were aligned, working with two maps each time, until all maps were fused into one. In [6] the proposed method outperforms the performance of the SOM in MSQE (Mean Squared Quantization Error) and topology preservation, by effectively locating the prototypes and relating the neighbor nodes. In [7] authors aimed to preserve map's topology in order to obtain the most truthful visualization of datasets. In [8] they investigate the use of relative validity indexes in cluster ensemble selection. These indexes select

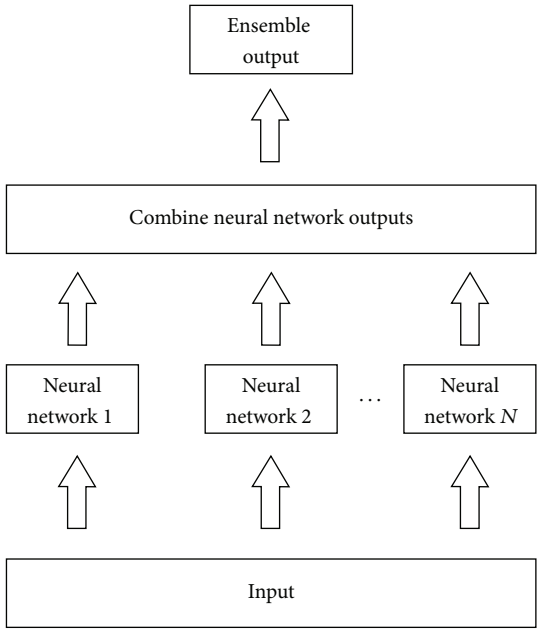


Figure 1: Block diagram of an ensemble.

the components with the highest quality to participate in the ensemble.

Different applications can be mentioned such as the mechanical fault diagnosis for high voltage circuit breakers [9], audio classification and segmentation [10], biomedical data [11], image segmentation [12], robotic [13], identification and characterization of computer attacks [14], unsupervised analysis of outliers on astronomical data [15], financial distress model [16], credit risk assessment [17], computer vision quality inspection of components [18], performance optimization of a classifier ensemble employed for target tracking in video sequences [19], disease prediction [20], and prediction of the duration of bus trips several days ahead [21]. Interesting reviews about this subject can be found in [22–25].

This work presents a contribution to the study of ensemble of Self-Organizing Maps with equal sizes. The goal is to improve the classification accuracy when compared to a single map. The paper is organized as follows: Section 2 presents some important concepts for this work, Section 3 shows the methodology proposed for the maps fusion, Section 4 presents the results obtained from the simulations, and Section 5 presents the conclusions and proposals for future works.

2. Background

2.1. Self-Organizing Maps. The Self-Organizing Maps (SOM) is a neural network model, known as a method for dimensionality reduction, data visualization, and data classification. It has competitive and unsupervised learning, performing a nonlinear projection of the input space R_e^p, with $p \gg 2$, in a grid of neurons arranged in two-dimensional array. This neural network has only two layers: an input and an output layer [4].

The network inputs x correspond to the p-dimensional vector space. The neuron i of the output layer is connected to all the inputs of the network, being represented by a vector of synaptic weights, also in p-dimensional space. Neurons in output layer are connected to the adjacent neurons through a neighborhood relationship that describes the topological structure of the map.

During the training phase, following a random sequence, input patterns x are compared to the neurons of the output layer. Through the Euclidean distance criterion a winning neuron, called BMU (Best Match Unit), is chosen and will represent the weight vector with the smallest distance to the input pattern; that is, the BMU will be the most similar to the input x. Assigning the winner neuron index by c, the BMU can be formally defined as a neuron according to

$$\|x - x_c\| = \arg\min_i \|x - w_i\|. \quad (1)$$

Equation (2) adjust the BMU weights and the neighboring neurons. Here t indicates the iteration of the training process, $x(t)$ is the input pattern, and $h_{ci}(t)$ is the nucleus of neighborhood around the winner neuron c:

$$w_i(t+1) = w_i(t) + \alpha(t) \cdot h_{ci}(t) \cdot [x(t) - w_i(t)]. \quad (2)$$

There are two ways of training a Kohonen map: sequential and batch training. The sequential training process of Self-Organizing Maps consists of three stages:

(1) Competition: the input pattern is presented to the network one by one and the output layer neurons will compete against each other. The smallest Euclidean distance between the synaptic weights of neurons and the input pattern determines the only winner (BMU).

(2) Cooperation: chosen the winner neuron, this becomes the center of a topological neighborhood of activated neurons.

(3) Adaptation: neurons of certain topological neighborhood adjusted their synaptic weights so that nearby neurons to BMU suffer greater adjustments than the more distant neurons.

Figure 2 shows the process of weights updating of winner neuron and its neighborhood. The solid lines correspond to the situation before the updating and the dashed lines show the change due to the approach of BMU and their input pattern neighbors x. The bright and dark circles, respectively, represent the neurons before and after adjustment of the weights.

The Gaussian function is used to describe the influence of the BMU on its neighbors (update the synaptic weights). The activated neuron tends to excite more intensively neurons in its immediate vicinity and decays smoothly with the lateral distance. Equation (3) shows how the neuron j excitation is calculated in relation to the winning neuron i where d_{ij} is the distance between the neurons i and j.

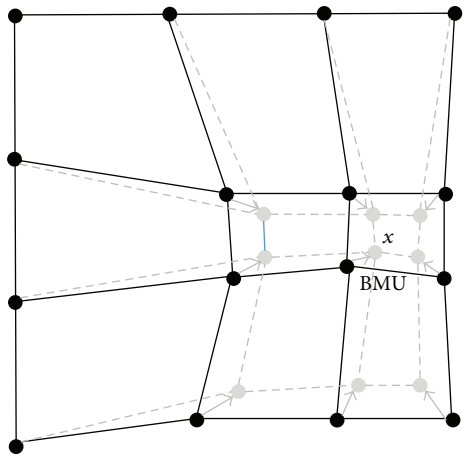


Figure 2: Weights updating [38].

At the end of the learning process, the output layer neurons represent a nonlinear approximation of the input patterns:

$$h_{ij} = \exp\left(-\frac{d_{ij}^2}{2\sigma^2}\right). \tag{3}$$

In batch training process all input patterns are presented to the network at once. Thus, the learning rate parameter $\alpha(t)$ is not necessary in (2) and randomness is not required in the presentation of the input vector to the network.

2.2. Cluster Validity Index. Cluster validity indexes (CVI) are used to evaluate the clustering algorithm results. Therefore, it is possible to check if an algorithm found the groups that best fit the data. The literature presents several CVI and most of them have high computational complexity, which can be a complicating factor in applications involving large data volumes. To overcome this problem, a modification in the CVI calculations was proposed in [16] using a vector quantization produced by Kohonen Map. The synaptic weight vectors (prototypes) are used instead of the original data and this leads to the decrease of the data amount, so the computational complexity for calculating the CVI decreases too. In the author's proposal, the hits should be part of the equation and not just prototypes, to avoid possible differences between the values calculated with all data and only the prototypes.

The CVI calculation always involves a measure of distance. The modification described in [26] changes the way these processes are performed. We explain this by comparing the two equations, the first for calculating the distance between two clusters C_i and C_j in the traditional way (4) and the other equation with the modifications for use with the SOM (5). One has

$$\delta_{i,j} = \frac{1}{|C_i|\,|C_j|} \sum_{\substack{z \in C_i \\ y \in C_j}} d(x, y). \tag{4}$$

In (4), $d(x, y)$ is a distance measure, and $|C_i|$ and $|C_j|$ refer to the clusters' amount of points C_i and C_j, respectively. When the amount of those points is high, the computational complexity is also high.

Equation (5) shows the modification considering the hits [26]:

$$\delta_{i,j} = \frac{1}{|C_i|\,|C_j|} \sum_{\substack{w_i \in W_i \\ w_j \in W_j}} h(w_i) \cdot h(w_j) \cdot d(w_i, w_j), \tag{5}$$

where W_i and W_j are the SOM prototype sets that represent the clusters C_i and C_j, respectively, $d(w_i, w_j)$ is the same distance measure type of (4), $h(w_i)$ is the prototype's hits w_i belonging to W_i, and $h(w_j)$ is the prototype's hits w_j belonging to W_j.

Equation (5) presents a lower computational cost since the quantities involved, w_i and w_j, are lower than C_i and C_j. The inclusion of the prototypes' hits leads to error minimization caused by the vector quantization that Kohonen Map produces, since it introduces in the calculation the approach for the points density in the input space, here represented by prototypes.

2.3. Bagging. There are three steps to obtain the components of an ensemble: generation, selection, and combination of individual components. In the component generation step it is necessary diversity among the components to ensure that each network generalizes differently. It can be achieved with random initialization of the weights for each component, the variation in the architecture of the neural network, the variation of the training algorithm or data resampling, among others.

In data resampling case, the Bagging [27] is a widely used technique. It consists in generating different training subsets from a single set via resampling with replacement. The samples selection from the original set has uniform probability distribution. Therefore, there will be some samples selected more than once, because of the replacement. There is the possibility that components do not generalize satisfactorily, but the ensemble of these models can result in a better generalization than each one of the individual components.

3. Materials and Methods

The method consists in the fusion of Kohonen maps formed from fractions of dataset created by bagging algorithm. The process overview is shown in Figure 3. The fusion occurs between two neurons that have the minimum Euclidean distance between them, indicating that they represent the same region of the input space.

The bagging method was used for generating subsets (resampling) from the training set, with uniform probability and replacement. The SOM was applied to each subset produced by bagging and all maps were segmented by k-means algorithm. Since not all the components should be used, because it can affect the final ensemble performance [28], the way to select these candidates will be through cluster validity index, modified for use with SOM, proposed by [26].

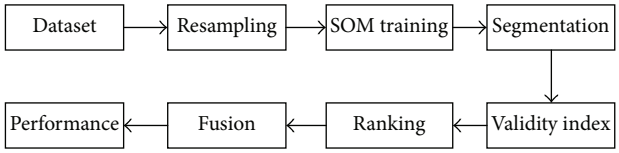


Figure 3: Equal sizes maps fusion process overview.

In this work the cluster validity indexes CDbw [29], Calinski-Harabasz [30], generalized Dunn [31], PBM [32], and Davies-Bouldin [33] were used as parameters in fusion decision process (consensus function).

The ranking-based selection method can be described in this way: the candidates to compose the ensemble are ranked based on clusters validity indexes values, which will measure the individual performance of each component.

3.1. Fusion Equations. We tested three different equations for neurons fusion. The first, equation (6), is an arithmetic average between the weight vectors of the neurons to be fused:

$$w_c = \frac{w_i + w_j}{2}. \tag{6}$$

The purpose of second equation was to avoid the influence of neurons without hits, as shown in

$$w_c = \frac{w_i \cdot h_i + w_j \cdot h_j}{h_i + h_j}. \tag{7}$$

Through (8), the third equation tested, we investigated whether the weighted fusion by CVI could result in an improvement in classification accuracy:

$$w_c = \frac{w_i \cdot h_i \cdot CV_i + w_j \cdot h_j \cdot CV_j}{h_i \cdot CV_i + h_j \cdot CV_j}, \tag{8}$$

where w_c is the fusion result of neurons w_i and w_j, considering h_i and h_j the prototype's hits and CV_i and CV_j the CVI from maps to be fused.

3.2. Factorial Experiment. We defined a factorial experiment to validate our approach. The maps size was varied in 10×10, 15×15, and 20×20. Through the bagging 10 sets were generated containing the following quantities of subsets: 10, 20, and 30 up to 100 subsets. These subsets were created with different data rates: 50%, 70%, and 90% of the training set; that is, for each map size, it has ten different numbers of subsets and three different values of bagging percentage. This combination results in 900 maps. Thus, the first experiment map has the size equal to 10×10, with 10 subsets generated by bagging with a percentage of 50%, as show in Figure 4.

3.3. Approaches. Four distinct approaches were defined to maps fusion, as a combination between ranked maps and maps fusion criteria. The ranking was based in five CVI and MSQE (Mean Square Quantization Error), defined in [6], and the maps fusion was based in five CVI and MSQE improvement criterion:

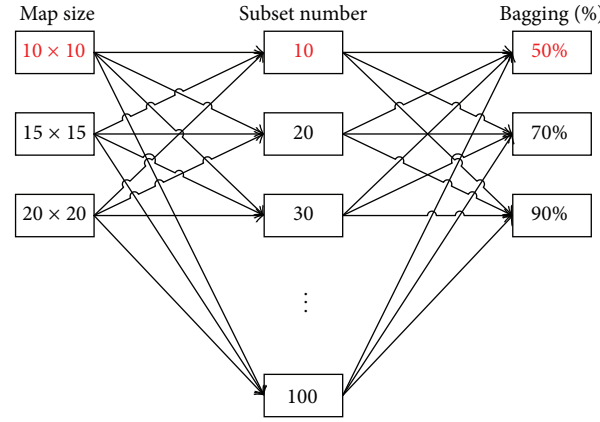


Figure 4: Structure of the factorial experiment.

(A) Maps ranked by CVI and fused by MSQE improvement criterion.

(B) Maps ranked by MSQE and fused by CVI improvement criterion.

(C) Maps ranked by CVI and fused by CVI improvement criterion.

(D) Maps ranked by MSQE and fused by MSQE improvement criterion.

The MSQE indicates how well the units of the map approximate the data on the dataset [7]; that is, it can be employed to evaluate the adaptation quality to the data [6].

In the first approach, the maps were ranked by five CVI, but the fusion process was controlled by MSQE improvement of the fused map. In the second approach, the maps were ranked by MSQE and the fusion process was validated by each one CVI improvement. In third approach, maps were ranked by each CVI and the fusion was controlled by CVI increase; that is, if the maps were ranked according to the Davies-Bouldin cluster validity index, the fusion will be controlled by the improvement of this index. At last, in the fourth approach the maps were ranked by MSQE and fused by MSQE improvement. Figure 5 shows the fusion process.

3.4. Proposed Fusion Algorithm. The proposed algorithm was applied in each one of the combinations shown in Figure 4 and according to approaches in Figure 5. It can be described as follows.

Algorithm 1 (overall fusion process).

Input. Dataset, map size, subset number, and bagging percentage.

(1) Generate, through the bagging, subsets from the training set.

(2) Apply the SOM to the subsets generated, resulting in N_i $(i = 1, 2, \ldots)$ maps with the same dimension.

(3) Segment the maps, with k-means algorithm.

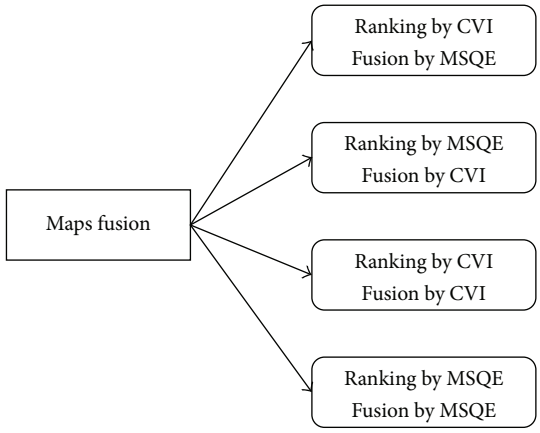


Figure 5: The four approaches to maps fusion.

(4) Calculate the CVI and MSQE for each N_i ($i = 1, 2, \ldots$) map and rank them according to this value, from the best index to the worst.

(5) The base map (one with the best CVI or MSQE value) is fused with the next best map, according to the sorted CVI or MSQE and (6), (7), and (8).

(6) Calculate the CVI or MSQE of fused map.

(7) If there is an improvement in the CVI or MSQE value, the map resulting from the fusion becomes the base map, then return to step (5) until all maps are fused in this way. Otherwise, discard the fusion between these two maps and return to step (5) until all maps are fused.

Output. Fused maps.

The main parameters of SOM were hexagonal lattice, Gaussian neighborhood, initial training radius equal to 4, and final training radius equal to 1 and 2000 training epochs.

Due to random initialization, k-means algorithm is repeated 10 times and the result with the smallest sum of squared errors is selected.

4. Results and Discussion

4.1. Datasets. The proposed method was tested through datasets with different characteristics from the UCI Repository [34] and from Fundamental Clustering Problems Suite (FCPS) [35], as shown in Table 1. Lines with missing values were removed from the datasets.

4.2. Results. The aim of the simulation was to verify in which map size the number of subsets and bagging percentage configuration best accuracy value could be obtained. Each experiment was run 10 times and mean values for a 95% confidential interval were obtained. The results were compared with a single SOM map, to verify the improvement in classification accuracy.

Table 1: Datasets.

Dataset	Repository	Instances	Attributes	Classes
Chainlink	FCPS	1000	3	3
Engytime	FCPS	4096	2	2
Lsun	FCPS	400	2	3
Tetra	FCPS	400	3	4
TwoDiamonds	FCPS	800	2	2
Wingnut	FCPS	1070	2	2
BC Wisconsin	UCI	699	9	2
Column	UCI	310	6	2
Heart	UCI	303	75	2
Ionosphere	UCI	351	34	2
Iris	UCI	150	4	3
PimaIndians	UCI	768	8	2
Seeds	UCI	210	7	3
Wine	UCI	178	33	3

4.2.1. General Result. Table 2 shows the best experimental results for each dataset; that is, which approach and which CVI resulted in the best accuracy value for each dataset.

The column "Equation" shows the best fusion equation for each dataset. Equation (6) is the simplest of the three tested equations, being just a simple average between two neurons weight vectors. Equation (7) is a weighted average of the hits neurons. Equation (8) makes a weighted average with hits and by CVI, the most complex equation. The results showed an equilibrium between (6) and (7) and (8) achieved the best accuracy for only two bases.

The column "Approach" refers to the way the maps were ranked and fused, as specified in Section 3.3. The best accuracy values were achieved with approaches B (maps ranked by MSQE and fused by CVI improvement criterion) and C (maps ranked by CVI and fused by CVI improvement criterion).

The column "CVI" shows which CVI was used to achieve the best fusion accuracy in relation to the single map, for each dataset (DB means Davies-Bouldin index and CH means Calinski and Harabasz index). The Davies-Bouldin index was the one that achieved the best fusion results over other CVI for these datasets and PBM index did not lead to good results.

The column "Map size" refers to the map size that has achieved the best accuracy value. The 15×15 map did not get the best results for these datasets.

The column "Bagging percentage" shows that, for the majority of dataset, using 90% of the samples for bagging produces better results. A high percentage of the bagging value has a wider range of data for training, leading to better fusion accuracy result. These results show that it is possible to get good accuracies without using the entire training set.

The column "Subsets number" shows the number of subsets that were evaluated to find the best accuracy. For most datasets used in this experiment the best accuracies were obtained with large amounts of subsets. This is an important conclusion: the number of maps available for fusing influences the accuracy that can be achieved.

Table 2: Best accuracies results.

Dataset	Equation	Approach	CVI	Map size	Bagging (%)	Subsets number	Single SOM accuracy	Proposed algorithm accuracy
BC Wisconsin	(6)	B	DB	20 × 20	90	80	0.9634	**0.9661**
Chainlink	(8)	B	DB	10 × 10	70	80	0.6654	**0.7685**
Column	(7)	C	DB	10 × 10	70	100	0.6674	**0.7219**
Engytime	(6)	A	CH	20 × 20	70	10	0.9538	**0.9587**
Heart	(6)	B	Dunn	20 × 20	90	30	0.7879	**0.8037**
Ionosphere	(7)	C	DB	10 × 10	90	20	0.7037	**0.7211**
Iris	(6)	B	Dunn	10 × 10	50	70	0.9200	**0.9260**
Lsun	(8)	C	DB	10 × 10	90	100	0.7500	**0.9850**
PimaIndians	(7)	C	CH	20 × 20	50	60	0.6458	**0.6939**
Seeds	(7)	C	DB	20 × 20	70	60	0.8809	**0.9243**
Tetra	(6)	B	Dunn	20 × 20	90	90	1	1
TwoDiamonds	(6)	C	Dunn	10 × 10	90	50	1	1
Wine	(6)	B	CDbw	10 × 10	90	80	0.9382	**0.9635**
Wingnut	(7)	C	CDbw	10 × 10	50	100	0.7539	**0.8983**

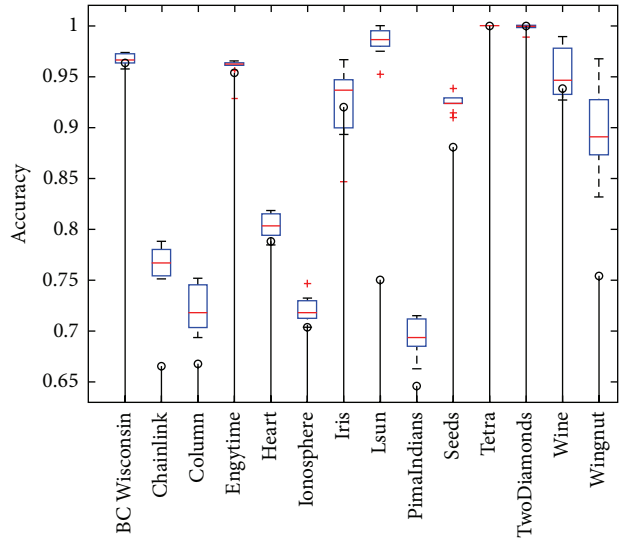


Figure 6: Classification accuracy for each dataset.

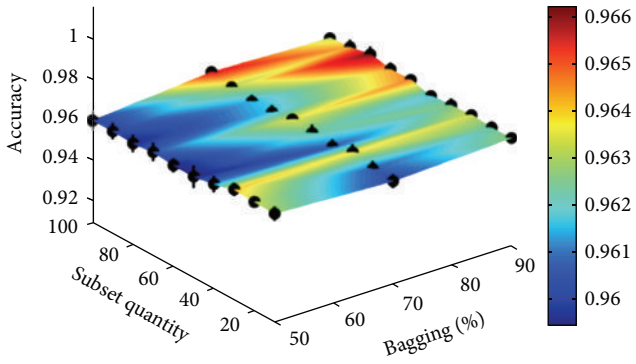


Figure 7: BC Wisconsin accuracy.

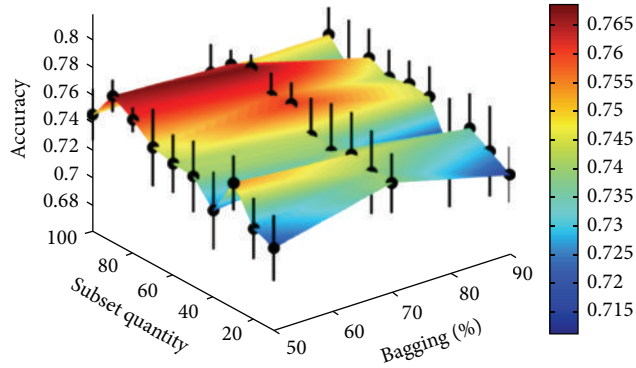


Figure 8: Chain-link accuracy.

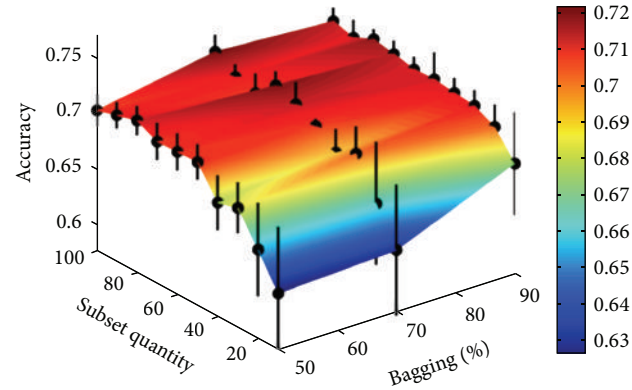


Figure 9: Column accuracy.

Figure 6 shows the classification accuracy obtained for each dataset. Black vertical lines show the single SOM accuracy value. As we can see, in all situations the accuracies achieved by the proposed method were higher than the classification accuracies for a single map, except for Tetra

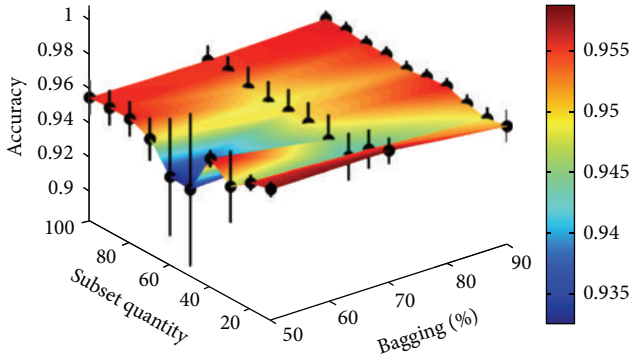


Figure 10: Engytime accuracy.

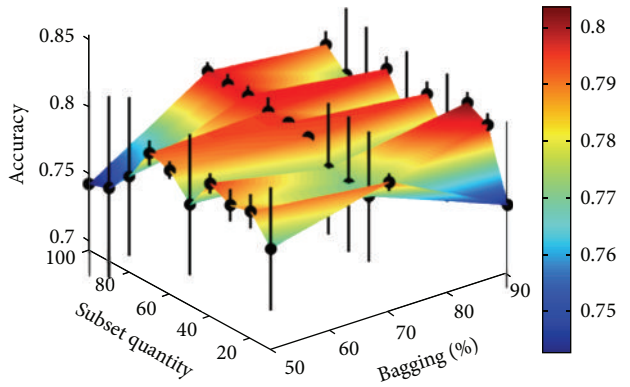


Figure 11: Heart accuracy.

and TwoDiamonds datasets. Thus, we show that through Self-Organizing Map fusion we can raise the value of classification accuracy.

In [36, 37] authors present ways to compare the performance of classifiers over multiple datasets. Among the tests presented in those papers, we chose to use the nonparametric Wilcoxon Signed Rank Test, at the significance level 0.05. This test was performed to evaluate whether the result obtained by Kohonen maps committee was statistically different from the accuracies obtained by a single SOM. The null hypothesis H_0 was that there is no difference between the accuracies obtained by the single map and the maps committee.

The Wilcoxon test rejected the null hypothesis, with p value equal to 0.0004882; that is, the proposed method obtained accurate results that are different and better than the accuracy of a single map.

4.2.2. Accuracy Results according to Bagging Percentage and Subset Quantity. Figures 7–20 show the accuracy (z-axis) obtained for each dataset according to bagging percentage (x-axis) and subsets quantity (y-axis) variation. Black lines represent the confidence interval for each experiment. In the 3D figures, hot regions represent the highest accuracy obtained and cold regions show lower accuracies obtained by the experiment. For example, Figure 7 shows that higher accuracies for BC Wisconsin dataset were achieved with

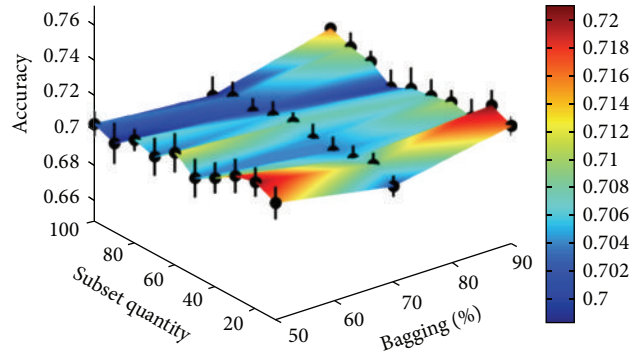


Figure 12: Ionosphere accuracy.

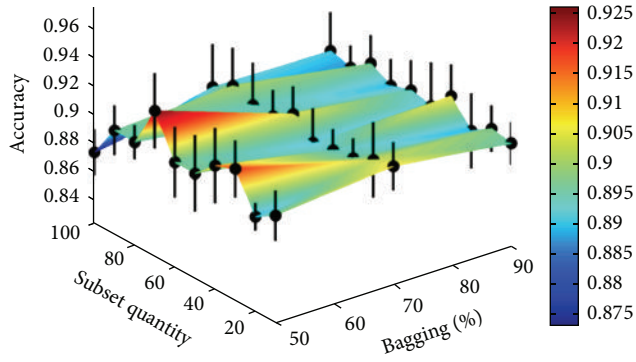


Figure 13: Iris accuracy.

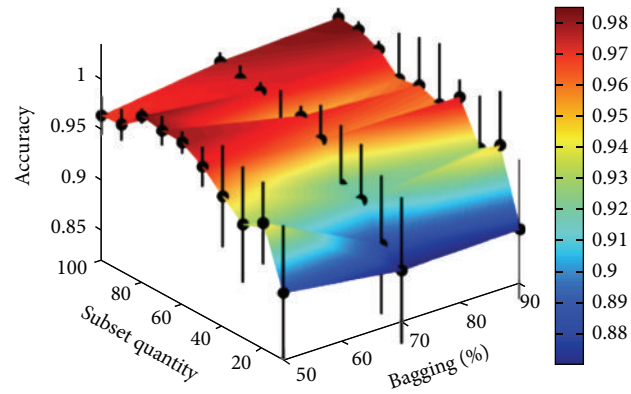


Figure 14: Lsun accuracy.

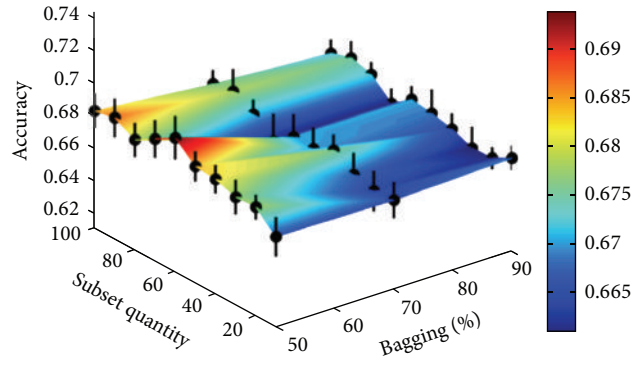


Figure 15: PimaIndians accuracy.

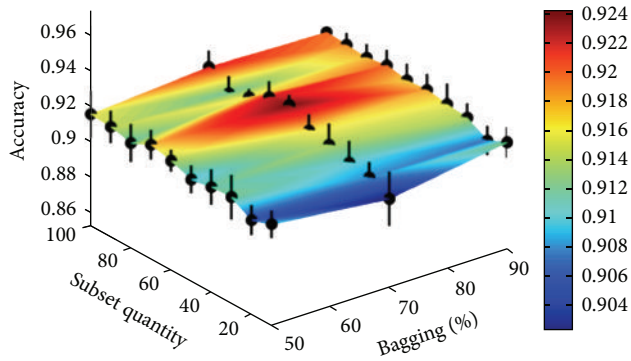


Figure 16: Seeds accuracy.

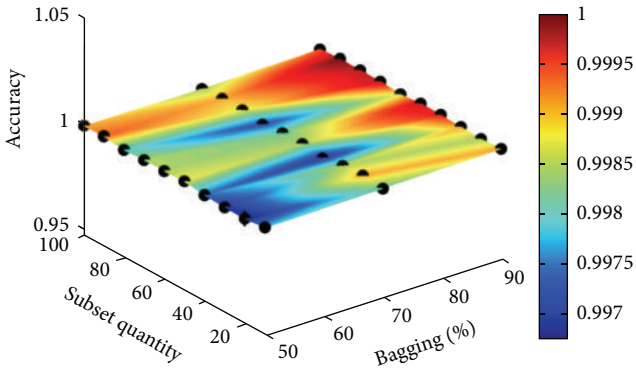


Figure 17: Tetra accuracy.

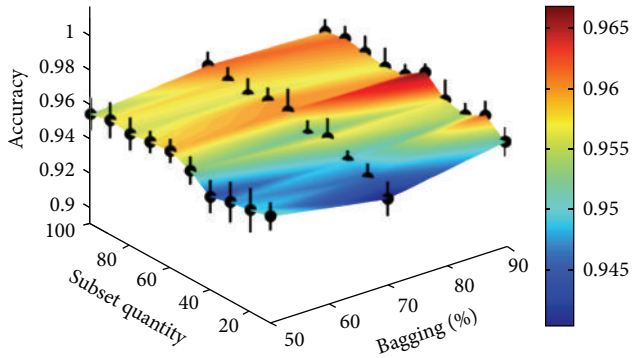


Figure 18: Wine accuracy.

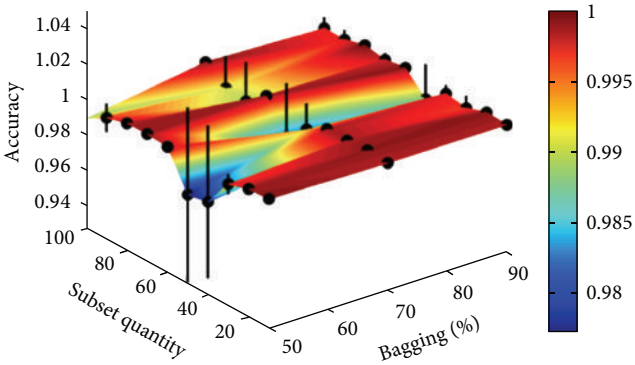


Figure 19: TwoDiamonds accuracy.

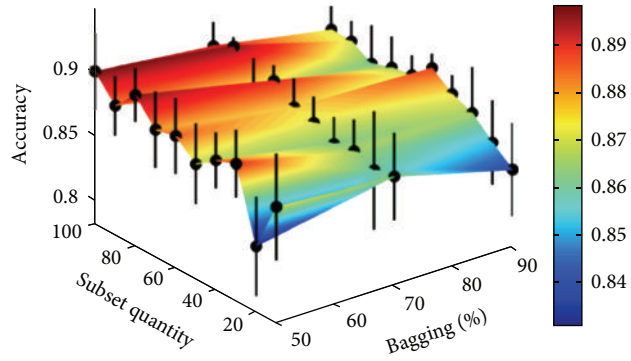


Figure 20: Wingnut accuracy.

a combination of higher amounts of subsets and a higher percentage of bagging.

5. Conclusions

This work investigated the possibility of fusing Kohonen maps ranked by cluster validity indexes and MSQE. Computer simulations evaluated different setting for map size, subsets number, and different amounts (percentage) of the training set. Datasets with distinct characteristics, including unbalanced classes, were selected to test the proposed method.

The Kohonen Map fusion method achieves values similar or superior to the ones observed in regular SOM (single map). Using a percentage of the training data (not all data), the proposed model achieved satisfactory results compared to SOM, which used all the training set.

The best results were obtained with high amounts of subsets and with high bagging percentage. Among the five CVI used in this study, only the PBM index did not lead to good results. Good results were achieved by ranking the maps by MSQE and fusing by CVI criterion and by ranking the maps by CVI and fusing by CVI criterion.

An important contribution of this paper is to show that we can use CVI and MSQE as a function consensus in ensemble of Self-Organizing Maps.

Future work may explore the influence of certain parameters such as the number of maps, the adjustments in SOM training, and the segmentation method and evaluate strategies to enhance visualization of the fused maps.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings*, vol. 1857 of *Lecture Notes in Computer Science*, pp. 1–15, Springer, Berlin, Germany, 2000.

[2] M. P. Perrone and L. N. Cooper, "When networks disagree: ensemble methods for hybrid neural networks," in *Neural Networks for Speech and Image Processing*, pp. 126–142, Chapman and Hall, 1993.

[3] J. F. Díez-Pastor, J. J. Rodríguez, C. I. García-Osorio, and L. I. Kuncheva, "Diversity techniques improve the performance of the best imbalance learning ensembles," *Information Sciences*, vol. 325, pp. 98–117, 2015.

[4] T. Kohonen, *Self-Organizing Maps*, vol. 30 of *Springer Series in Information Sciences*, Springer, Berlin, Germany, 2nd edition, 1997.

[5] A. Georgakis, H. Li, and M. Gordan, "An ensemble of SOM networks for document organization and retrieval," in *Proceedings of the International Conference on Adaptive Knowledge Representation and Reasoning (AKRR '05)*, Espoo, Finland, 2005.

[6] C. Sandoval, R. Salas, S. Moreno, and H. Allende, "Fusion of self-organizing maps," in *Computational and Ambient Intelligence*, vol. 4507 of *Lecture Notes in Computer Science*, pp. 227–234, Springer, Berlin, Germany, 2007.

[7] E. Corchado and B. Baruque, "WeVoS-ViSOM: an ensemble summarization algorithm for enhanced data visualization," *Neurocomputing*, vol. 75, pp. 171–184, 2012.

[8] M. C. Naldi, A. C. P. L. F. Carvalho, and R. J. G. B. Campello, "Cluster ensemble selection based on relative validity indexes," *Data Mining and Knowledge Discovery*, vol. 27, no. 2, pp. 259–289, 2013.

[9] J. Zhang, M. Liu, K. Wang, and L. Sun, "Mechanical fault diagnosis for HV circuit breakers based on ensemble empirical mode decomposition energy entropy and support vector machine," *Mathematical Problems in Engineering*, vol. 2015, Article ID 101757, 6 pages, 2015.

[10] S. Zahid, F. Hussain, M. Rashid, M. H. Yousaf, and H. A. Habib, "Optimized audio classification and segmentation algorithm by using ensemble methods," *Mathematical Problems in Engineering*, vol. 2015, Article ID 209814, 11 pages, 2015.

[11] S. J. Fodeh, C. Brandt, T. B. Luong et al., "Complementary ensemble clustering of biomedical data," *Journal of Biomedical Informatics*, vol. 46, no. 3, pp. 436–443, 2013.

[12] Y. Jiang and Z.-H. Zhou, "SOM ensemble-based image segmentation," *Neural Processing Letters*, vol. 20, no. 3, pp. 171–178, 2004.

[13] K. H. Low, W. K. Leow, and M. H. Ang Jr., "An ensemble of cooperative extended kohonen maps for complex robot motion tasks," *Neural Computation*, vol. 17, no. 6, pp. 1411–1445, 2005.

[14] L. L. DeLooze, "Attack characterization and intrusion detection using an ensemble of self-organizing maps," in *Proceedings of the IEEE Information Assurance Workshop*, pp. 108–115, IEEE, West Point, NY, USA, June 2006.

[15] D. Fustes, C. Dafonte, B. Arcay et al., "SOM ensemble for unsupervised outlier analysis: application to outlier identification in the Gaia astronomical survey," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1530–1541, 2013.

[16] C.-F. Tsai, "Combining cluster analysis with classifier ensembles to predict financial distress," *Information Fusion*, vol. 16, pp. 46–58, 2014.

[17] R. F. Lopez and J. M. R. Jeronimo, "Enhancing accuracy and interpretability of ensemble strategies in credit risk assessment. A correlated-adjusted decision forest proposal," *Expert Systems with Applications*, vol. 42, no. 13, pp. 5737–5753, 2015.

[18] S. K. Goumas, I. N. Dimou, and M. E. Zervakis, "Combination of multiple classifiers for post-placement quality inspection of components: a comparative study," *Information Fusion*, vol. 11, no. 2, pp. 149–162, 2010.

[19] I. Visentini, L. Snidaro, and G. L. Foresti, "Diversity-aware classifier ensemble selection via f-score," *Information Fusion*, vol. 28, pp. 24–43, 2016.

[20] C. H. Weng, T. C. Huang, and R. P. Han, "Disease prediction with different types of neural network classifiers," *Telematics and Informatics*, vol. 33, no. 2, pp. 277–292, 2016.

[21] J. M. Moreira, A. M. Jorge, J. F. Sousa, and C. Soares, "Improving the accuracy of long-term travel time prediction using heterogeneous ensembles," *Neurocomputing*, no. 150, pp. 428–439, 2015.

[22] N. C. Oza and K. Tumer, "Classifier ensembles: select real-world applications," *Information Fusion*, vol. 9, no. 1, pp. 4–20, 2008.

[23] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognition*, vol. 44, no. 8, pp. 1761–1776, 2011.

[24] S. Vega-Pons and J. Ruiz-Shulcloper, "A survey of clustering ensemble algorithms," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 3, pp. 337–372, 2011.

[25] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Information Fusion*, vol. 16, no. 1, pp. 3–17, 2014.

[26] M. L. Gonçalves, M. L. De Andrade, J. A. Ferreira Costa, and J. Zullo, "Data clustering using self-organizing maps segmented by mathematic morphology and simplified cluster validity indexes: an application in remotely sensed images," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '06)*, pp. 4421–4428, IEEE, Vancouver, Canada, July 2006.

[27] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[28] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: many could be better than all," *Artificial Intelligence*, vol. 137, no. 1-2, pp. 239–263, 2002.

[29] M. Halkidi and M. Vazirgiannis, "A density-based cluster validity approach using multi-representatives," *Pattern Recognition Letters*, vol. 29, no. 6, pp. 773–786, 2008.

[30] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, no. 2, pp. 159–179, 1985.

[31] J. C. Bezdek and N. R. Pal, "Some new indexes of cluster validity," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 28, no. 3, pp. 301–315, 1998.

[32] M. K. Pakhira, S. Bandyopadhyay, and U. Maulik, "Validity index for crisp and fuzzy clusters," *Pattern Recognition*, vol. 37, no. 3, pp. 487–501, 2004.

[33] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 224–227, 1979.

[34] K. Bache and M. Lichman, *Machine Learning Repository*, School of Information and Computer Sciences, University of California, Irvine, Irvine, Calif, USA, 2013, http://archive.ics.uci.edu/ml.

[35] A. Ultsch, "Clustering with SOM. U*C," in *Proceedings of the Workshop on Self-Organizing Maps (WSOM '05)*, pp. 75–82, Paris, France, 2005.

[36] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[37] S. García and F. Herrera, "An extension on 'statistical comparisons of classifiers over multiple data sets' for all pairwise comparisons," *Journal of Machine Learning Research*, vol. 9, pp. 2677–2694, 2008.

[38] J. Vesanto, *Data exploration process based on the self-organizing map [Ph.D. thesis]*, Helsinki University of Techonology, 2002.