Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2013, Article ID 589386, 6 pages
http://dx.doi.org/10.1155/2013/589386

Research Article

An Enhanced Wu-Huberman Algorithm with Pole Point Selection Strategy

Yan Sun[1] and Shuxue Ding[2]

[1] *School of Psychology, Liaoning Normal University, Dalian 116029, China*
[2] *School of Computer Science and Engineering, Aizu University, Aizuwakamatsu 965-8580, Japan*

Correspondence should be addressed to Yan Sun; sunyan@lnnu.edu.cn

Received 26 February 2013; Accepted 23 April 2013

Academic Editor: Fuding Xie

The Wu-Huberman clustering is a typical linear algorithm among many clustering algorithms, which illustrates data points relationship as an artificial "circuit" and then applies the Kirchhoff equations to get the voltage value on the complex circuit. However, the performance of the algorithm is crucially dependent on the selection of pole points. In this paper, we present a novel pole point selection strategy for the Wu-Huberman algorithm (named as PSWH algorithm), which aims at preserving the merit and increasing the robustness of the algorithm. The pole point selection strategy is proposed to filter the pole point by introducing sparse rate. Experiments results demonstrate that the PSWH algorithm is significantly improved in clustering accuracy and efficiency compared with the original Wu-Huberman algorithm.

1. Introduction

Traditional data mining approaches can be categorized into two categories [1]: one is supervised learning, which aims to predict the labels of any new data points from the observed data-label pairs. Typical supervised learning methods include the support vector machine and the decision trees; the other one is unsupervised learning. The goal is just to organize the observed data points with no labels. Typical unsupervised learning tasks include clustering [2] and dimensionality reduction [3]. In this paper, we will focus on the clustering problem, which aims to divide data into groups with similar objects. From a machine learning perspective, clustering is to learn the hidden patterns of the dataset in an unsupervised way. From a practical perspective, clustering plays a vital role in data mining applications such as information retrieval, text mining, web analysis, marketing, and computing biology [4–7].

In the last decades, many methods [8–12] have been proposed for clustering. Recently, the graph-based clustering has attracted many interests in the machine learning and data mining community [13]. The cluster assignments of the dataset can be achieved by optimizing some criteria defined on the graph. For example, the spectral clustering is one kind of the most representative graph-based clustering approaches, and it aims to optimize some cut values (e.g., [14, 15]) defined on an undirected graph. After some relaxations, these criteria can usually be optimized via eigen decompositions, and the solutions are guaranteed to be globally optimal. In this way, the spectral clustering efficiently avoids the problems of the traditional K-means method.

Wu and Huberman proposed a clustering method based on the notation of voltage drops across the network [16]. The algorithm uses a statistical method to avoid the "poles problem" instead of solving it. The idea randomly picks two poles, then applies the algorithm to divide the graph into two communities, and repeats in this way for many times. The algorithm uses a majority vote to determine the communities [16]. However, after making some experiments, we have found that the choice of the pole points affects the accuracy of some of the clustering so seriously that the majority voting result is degraded. The specific details will be presented in Section 4.1 (Figure 1).

In order to overcome the above disadvantages of the Wu-Huberman algorithm, in this paper, first we construct a graph in terms of data points. Then we propose a novel strategy

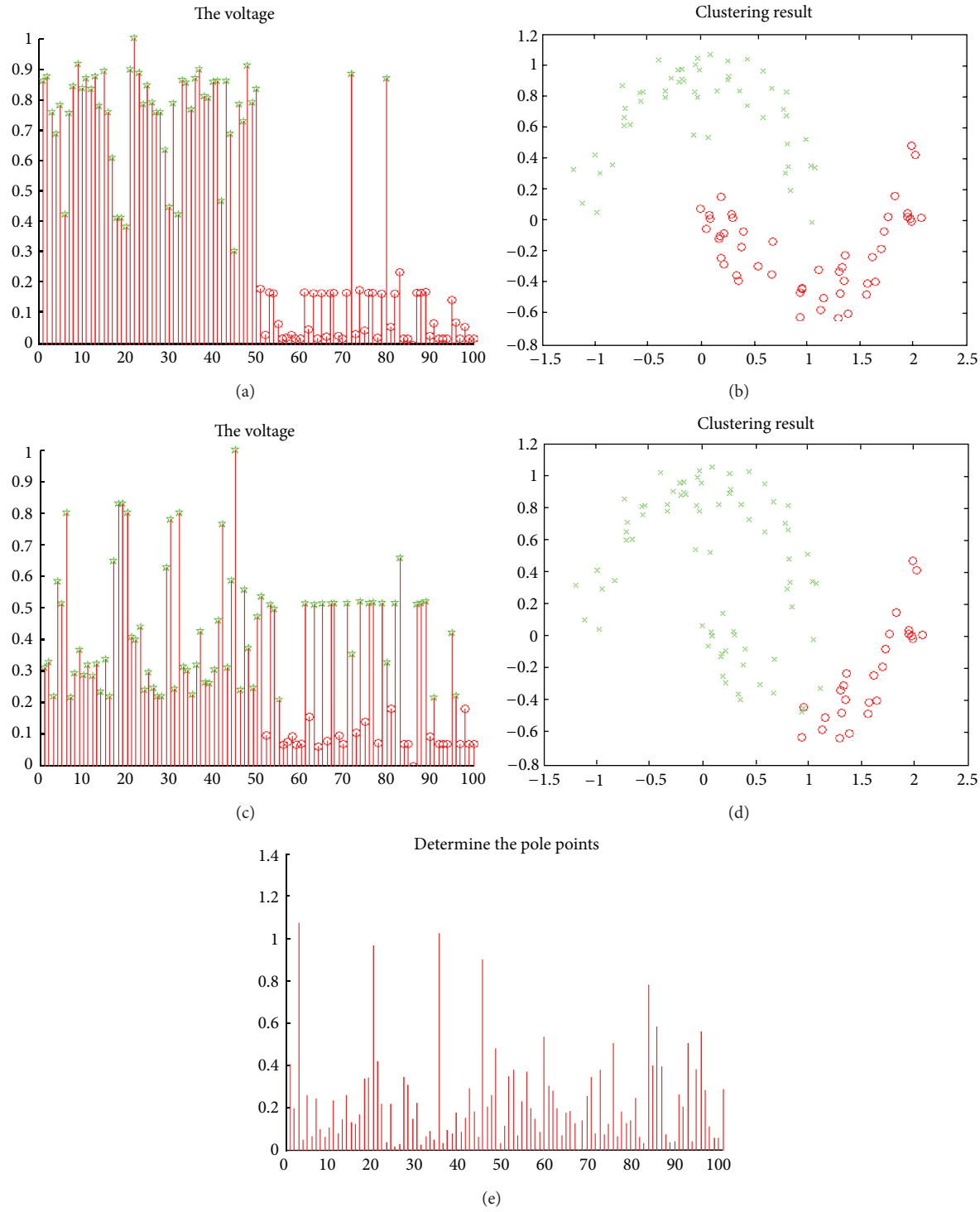

FIGURE 1: Clustering results of the Wu-Huberman algorithm for the two-moon pattern with different pole point selections. (a) The distribution of the voltage values when 22nd and 86th points have been chosen as the poles. (b) The clustering results corresponding with (a). (c) The distribution of voltage values when the 45th and 86th points have been chosen as the poles under the same dataset, algorithm, and parameters with (a). (d) The clustering results corresponding to (c). (e) The graph of determining the pole points. The x-axis is the data point number, and the y-axis is the value of sparse rate δ.

for pole point selection. After that, we iteratively solve the Kirchhoff equation to perform clustering. Finally, we get the clustering result. In this paper, we consider only the 2-community clustering case and will leave the case of k cluster problem into the future research.

2. Related Works

The Wu-Huberman algorithm exhibits the graph as an electric circuit. The purpose is to classify points in the graph into two communities, that is, clusters. We denote a graph by $G = (X, E)$, where X is the point set of graph and E is the edge set. The set of voltages of points is V. Suppose points A and B have been known to belong to different communities, G_1 and G_2, respectively. By solving Kirchhoff equations the voltage value of each point can be obtained, which of course should lie between 0 and 1. A point belongs to G_1 or G_2, which can be decided by voltage value of the point [17]. The graph is regarded as an electric circuit by associating a unit resistance to each of its edges. Two of the nodes, assumed to be node 1 and node 2, without losing the generality, in the graph are given a fixed potential difference. The Wu-Huberman method is based on an approximate iterative algorithm that solves the Kirchhoff equations for node voltages in linear time [16, 18].

The Kirchhoff equations of n-point circuit can be written as

$$V_1 = 1, \qquad V_2 = 0,$$
$$V_i = \frac{1}{d_i} \sum_{(i,j)\in E} V_j = \frac{1}{d_i} \sum_{j\in G} V_j a_{ij}, \quad \text{for } i = 3, \ldots, n, \tag{1}$$

where d_i is the degree of point x_i and a_{ij} is the adjacency matrix of the graph. After the convergence, each community, that is, cluster, is defined as the nodes with a specific voltage value within a tolerance. Without loss of generality, the algorithm has labeled the point in such a way that the battery is attached to point 1 and 2, which are termed as pole points.

Because of the complexity, the algorithm does not solve the Kirchhoff equations exactly rather solves it iteratively. The algorithm initially sets $V_1 = 1, V_2 = \cdots = V_n = 0$. In the first round, the algorithm starts updating from point 3 to the nth point in the following way. When the ith point, the voltage of it is substituted by the average value of its k neighbors according to (1). The updating process ends when the algorithm gets to the last point n, at which a *round* is finished. After repeating the updating process for a finite number of rounds, each point reaches voltage value that satisfies approximately the Kirchhoff equations within a certain precision. Then the algorithm finds community results by a threshold decision.

The Wu-Huberman algorithm inherits the superiority of the graph-based clustering. The final cluster solutions is global optimal. Especially, the running time of the algorithm is linear. However, the algorithm does not always work in many cases [16]. Besides, there is still one critical problem which seriously affects the accuracy and efficiency in real applications. That is, the accuracy and efficiency are greatly affected by the poles, that is, node 1 and node 2 selected. Therefore, it is most important to improve the method of selecting poles. In this paper, we present the PSWH algorithm to improve the accuracy and effectiveness of the algorithm by presenting the pole point selection strategy.

3. The PSWH Algorithm

3.1. Graph Construction. Let $G = (X, E)$ be an undirected graph with point set $X = \{x_1, \ldots, x_n\}$ and edge set $E \subseteq X \times X$. The degree of point $x_i \in X$ is defined as d_i, which is the edge number connecting with point x_i.

Constructing k nearest neighborhood graph is to model the local neighborhood relationships between the data points. Given data points $x_1, \ldots, x_n$, we link x_i and x_j with an undirected edge if x_i is among the k nearest neighbors of x_j or if x_j is among the k nearest neighbors of x_i. We define x_i and x_j to be adjacent if $x_i \in N(x_j)$ or $x_j \in N(x_i)$, $N(x_i)$, and $N(x_j)$ is the neighbor of x_i and x_j, respectively. w_{ij} is the similarity between x_i and x_j. w_{ij} is computed in the following way: $w_{ij} = e^{-(\|x_i - x_j\|^2/2\sigma^2)}$, where σ is a dataset-dependent parameter.

3.2. The Pole Point Selection Strategy. The Wu-Huberman algorithm selects pole point randomly. Based on plenty of experiments, we find that clustering results are very sensitive to the choosing of pole points. It may produce wrong clustering results if inappropriate points are chosen as the poles. Figure 1 gives us an intuitive illustration of such a problem.

For solving this problem, in this paper, we introduce a concept that is termed as "sparse points." There is the maximal diameter between the sparse point and its neighborhoods. The existence of sparse points will bias the final clustering results. An important fact of our experimental results is that if we choose sparse points as the pole points the Wu-Huberman algorithm will become less accurate. For this reason, the sparse points should not be selected as the pole points. Therefore, we propose the following sparse rate δ_i to discriminate the sparse points from the others. Additionally, in order to exclude the impact of the distribution in the similarity and degree, the averaging similarity of the neighbors and the similarity summation of the neighbors should be taken in the sparse rate δ_i. That is,

$$\delta_i = \frac{\gamma_i}{\left(\overline{\lambda}_i \times \lambda_i\right)}, \tag{2}$$

where γ_i is the maximum diameter between the ith point and its neighborhoods; $\gamma_i = \max \arg \sqrt{\sum_{j=1,p=1}^{d_i} \sum_{q=1}^{\text{number\_}f} (x_{ijq} - x_{ipq})^2}$, $i = 1$ to n, x_{ij} and x_{ip} are the neighborhoods of the x_i, j and p are from 1 to d_i, number_f is the feature number of x_i, and x_{ijq} is the qth attribute feature in the jth neighborhood of x_i. Here λ_i is the similarity (weight) summation of x_i's neighborhood, $\lambda_i = \sum_{j=1}^{d_i} w_{ij}$, $i = 1$ to n. $\overline{\lambda}_i$ is the average weight of x_i's neighborhood, $\overline{\lambda}_i = \lambda_i / d_i$.

Figure 1(e) shows the sparse rate of each point in Figure 1. A point can be determined as the pole point whose sparse

rate is significantly larger than those of the most other points. Sparse points are far from other points between two different clusters, so they should not be chosen as the pole points.

We define an extent to describe the range of allowed sparse points' number. For example, an extent of 5% in the two-moon example means that the allowed sparse point number is the number of points $*$ extent $= 100 * 5\% = 5$. That is to say, we choose top 5 points upon the sparse rate as the sparse points. The specific experimental details are shown in Section 4.1.

3.3. Iteratively Solving the Kirchhoff Equations. We will illustrate the computation procedure for iteratively solving the Kirchhoff equations by using an example. According to the results of (2), we get that the pole points are 1st and nth points. That is to say, $v_1 = 1, v_2 = \cdots = v_n = 0$. Then use (1) to obtain the voltage value of each point excluding the pole points, at which the voltage values are fixed. That is, the value of each point is the similarity average of its neighbor point. The updating process ends when we go through 2th to n-1th points. Repeat this process till voltage value converges within stable error range. In our experiments, we set 0.001 as the terminative conditioning of the iteration.

3.4. The Procedure of the PSWH Algorithm

Input. Dataset $X = \{x_i\}_{i=1}^n$ and the neighborhood size k.

Output. The cluster membership of each data point.

Procedure

Step 1: construct the k nearest neighborhood.

Step 2: compute sparse rate δ_i using (2) and apply the extent to determine the pole points. Then exclude the sparse points in graph and choose randomly two other points as the pole points.

Step 3: obtain the voltage value of each data point based on (1).

Step 4: output the cluster assignments of each data point.

4. Experimental Results

In this section, we will use the well-known two-moon example to illustrate the effectiveness of PSWH algorithm. The original dataset is a standard benchmark for machine learning algorithms [19] and is generated according to a pattern of two intertwining crescent moons. This benchmark is online available at http://www.ml.uni-saarland.de/GraphDemo/GraphDemo.html. In the experiments, the Gaussian noise with mean 0 and variance 0.01 has been added. The number of data points is set as 100 for the two moons.

4.1. Pole Points' Influence on the Clustering Accuracy. In the Wu-Huberman algorithm, the choice of the pole points affects significantly the clustering results. Taking the two-moon dataset as an example, we set σ as 0.5 and k as 5. In Figure 1(e), the sparse points are the 3rd, 20th, 35th, 45^{th}, and 83rd points. In order to improve the clustering accuracy, we do not choose the sparse points as the poles. The clustering accuracy is 100%. Figure 1(c) illustrates that no matter what threshold is chosen, the cluster accuracy is low. That is to say, the choice of the poles has great effect on the clustering results.

4.2. Pole Points' Influence on the Iterate Number. In the experiment, we find that the choice of the pole points has an impact on the iterate number. The two-moon dataset is taken as an example. All of the experiments are conducted in the same parameter conditions: such as $\sigma = 0.5$, the iterate error is 0.001, and the maximum iterate number is 100.

We first construct the KNN ($k = 5$) graph of original dataset. Then the degree of each point was computed and displayed in Figure 2(b). Next, we obtain the sparse rate of each point based on the degree distribution, which is the same as Figure 1(e). Finally, we choose the poles based on the sparse rate, compute (1) to obtain the voltage value of each point, and, respectively, display the iterate number of each point in Figures 2(c) and 2(d) when different poles are chosen.

In Figure 2, we can draw a conclusion that the greater degree of the poles corresponds to the more iterate number for convergence. Therefore, in order to decrease the iterate number of the algorithm, we should choose the points with smaller sparse degree as the poles. The clustering accuracy of Figure 2 is 100%.

4.3. Comparison with Other Algorithms. We compare the PSWH algorithm with other algorithms on the UCI repository, which is available at http://archive.ics.uci.edu/ml/.

From Table 1, we can find that the PSWH algorithm does slightly better than other algorithms in most dataset. However, in some conditions, the PSWH algorithm is lower than LCLGR algorithm. Considering the complexity of algorithm is linear, which is lower than LCLGR algorithm. Therefore, in general, the PSWH algorithm is an excellent algorithm than the others.

5. Conclusions and Future Work

In this paper, we propose PSWH algorithm for enhancing the clustering accuracy and efficiency of the Wu-Huberman algorithm, which can extend the applicability and increase the robustness of the algorithm. The concept of sparse points and selection procedure are presented to obtain the suitable pole points for the algorithm. The experimental results showed that the PSWH algorithm is very effective and stable when applied to clustering problems. In the future, we will give the theoretical analysis of the new algorithm and employ the new algorithm to more general and larger datasets. Furthermore, we will try to extend the new algorithm to textual, image, and video retrievals.

Table 1: Comparison with other algorithms on the clustering accuracy.

Data sets	K-means	Ncut [15]	LCLGR [1]	Wu-Huberman [16]	PSWH
BUPA	0.5623	0.5710	0.6493	0.5304	0.6145
Balance	0.5472	0.5195	0.5664	0.9983	0.9983
Monks	0.5806	0.7097	0.7339	0.6452	0.6690
Iris	0.5533	0.9867	0.9933	1	1
Crx	0.5038	0.6677	0.7871	0.5758	0.6263
Wine	0.5000	0.7416	0.8371	0.6667	0.7536
Hayes-Roth	0.4242	0.4015	0.4394	0.4015	0.4318

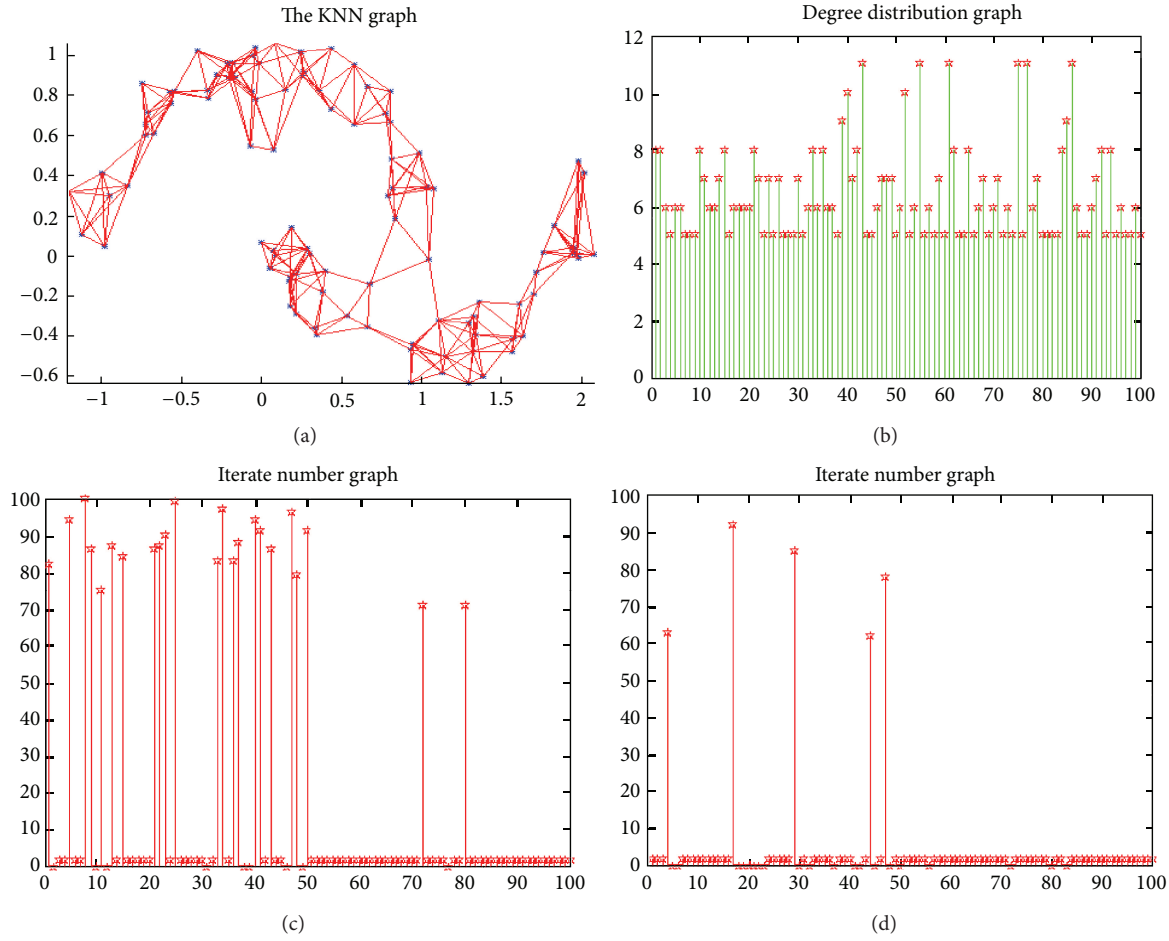

Figure 2: Different pole points of the Wu-Huberman algorithm were applied, which leads to different iterate number of convergence. (a) The KNN ($k = 5$) graph. (b) The degree distribution graph. (c) The iterate number via vertical axis when the poles are the 2nd point (its degree is 8) and the 77th point (its degree is 11). (d) The iterate number via vertical axis when the poles are the 5th point (its degree is 6) and the 56th point (its degree is 5), where the x-axis represents the data points and y-axis represents the iterate number.

Acknowledgments

This work was supported by the key project of the National Social Science Fund (11AZD089) and Educational Commission Scientific Project of Liaoning Province (no. L2012381).

References

[1] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 55–67, 2008.

[2] Y. Sun, Y. Y. Tang, and L. Z. Yang, "An adaptive selection strategy and separation measure for improving the Wu-Huberman clustering," *ICIC Express Letters B*, vol. 3, no. 6, pp. 1531–1536, 2012.

[3] L. K. Saul, K. Q. Weinberger, F. Sha, J. Ham, and D. D. Lee, *Spectral Methods for Dimensionality Reduction*, MIT Press, 2006.

[4] W. H. Cui, W. Wang, X. B. Liu, and J. S. Wang, "An improved clustering algorithm for product family design of paper currency sorter," *IICIC Express Letters B*, vol. 3, no. 4, pp. 909–915, 2012.

[5] C. Cheng, D. Zhang, Z. Yu, and H. Li, "High speed data streams clustering algorithm based on improved SS tree," *ICIC Express Letters B*, vol. 3, no. 1, pp. 207–212, 2012.

[6] F. Wang, C. Zhang, and T. Li, "Clustering with local and global regularization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 12, pp. 1665–1678, 2009.

[7] X. Wang, X. Wang, and D. M. Wilkes, "A divide-and-conquer approach for minimum spanning tree-based clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 7, pp. 945–958, 2009.

[8] D. X. Chang, X. D. Zhang, and C. W. Zheng, "A genetic algorithm with gene rearrangement for K-means clustering," *Pattern Recognition*, vol. 42, no. 7, pp. 1210–1222, 2009.

[9] L. Jing, M. K. Ng, and J. Z. Huang, "An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 8, pp. 1026–1041, 2007.

[10] M. K. Ng, M. J. Li, J. Z. Huang, and Z. He, "On the impact of dissimilarity measure in κ-modes clustering algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 503–507, 2007.

[11] M. H. Wang, Y. F. Tseng, H. C. Chen, and K. H. Chao, "A novel clustering algorithm based on the extension theory and genetic algorithm," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8269–8276, 2009.

[12] K. R. Žalik, "An efficient k'-means clustering algorithm," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1385–1391, 2008.

[13] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: a geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol. 7, no. 11, pp. 2399–2434, 2006.

[14] C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A min-max cult algorithm for graph partitioning and data clustering," in *Proceedings of the 1st IEEE International Conference on Data Mining (ICDM '01)*, pp. 107–114, San Jose, Calif, USA, December 2001.

[15] 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.

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

[17] S. E. Schaeffer, "Graph clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27–64, 2007.

[18] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: structure and dynamics," *Physics Reports*, vol. 424, no. 4-5, pp. 175–308, 2006.

[19] O. Chapelle, V. Sindhwani, and S. Keerthi, "Branch and bound for semi-supervised support vector machines," in *Advances in Neural Information Processing Systems*, B. Scholkopf, J. Platt, and T. Hoffman, Eds., vol. 19, MIT Press, Cambridge, Mass, USA, 2007.