International Journal of Database Theory and Application
Vol.8, No.6 (2015), pp.39-46
http://dx.doi.org/10.14257/ijdta.2015.8.6.04

Applying Bi-clustering Algorithm in Customer Segmentation for High-Value Customers

Wang Chengduan

School of computer engineering, Weifang University, 261061, Weifang, China
wangchengduan@163.com

Abstract

As one of the most popular data mining techniques, clustering is an important way of exploratory data analysis and pattern discovery given the explosive increasing amount of dataset. Applying clustering in customer segmentation is a common method to discover high-value customers. However, traditional clustering methods such as k-means are performing single direction (either row or column) on the data matrix, and thus the clustering results might involve cases that are irrelevant to specific dimensions. Besides, traditional clustering is achieved upon the whole set of attributes or variables, and therefore only capable of discovering global information. Along this line, in order to reduce the dimensions and find out potential local patterns in the data matrix, we proposed a bi-clustering algorithm for customer segmentation. Our experiments using supermarket customer dataset improve the effectiveness and efficiency of proposed bi-clustering algorithm.

Keywords: *Clustering, Bi-clustering, Customer segmentation, High-value customer discovery*

1. Introduction

The explosive increasing amount of data makes it even more difficult to discover meaningful information, and therefore how to acquire information effectively and efficiently has become one of the most significant focuses. That is what data mining techniques are used for. As an interdisciplinary field, data mining [1] refers to the process and techniques involved to find out useful information from a pile of data, and has been successfully applied in many areas [2-5].

As one of the most popular data mining techniques, clustering is an important way of exploratory data analysis and pattern discovery. The basic idea of clustering is to group the dataset into several clusters, such that the objects within the same cluster are similar enough and the objects across different clusters are different enough [1]. Many clustering methods have been proposed, such as hierarchical clustering [6], k-means [7], Self-Organizing Maps (SOM) [8], and Support Vector Machines (SVM) [9], *etc.*

However, traditional clustering methods are performing single direction (either row or column) on the data matrix, and thus the clustering results might involve cases that are unrelated to specific dimensions. Besides, traditional clustering is achieved upon the whole set of attributes or variables, and therefore only capable of discovering global information. Indeed, given the rapidly growing of dataset scale and dimension, in order to reduce the dimensions and find out potential local patterns in the data matrix, bi-clustering was proposed [10] and has been widely applied in microarray bioinformatics [11], text mining [12], and recommender systems [13], *etc.*

Customer segmentation in one of the most significant concept in Customer Relationship Management (CRM) [14], which refers to the process of dividing customers into groups of individuals with similarity in terms of specific attributes, such as age, gender, credit, *etc.* According to the 80-20 rules [15] in CRM, that is, 80% profit comes

ISSN: 2005-4270 IJDTA

from 20% customers, discovering high potential customers and maintaining relationships with customers is the key in market competition for enterprises. By precisely identifying different segments of customer base and customizing products and services for different groups of customers, differentiated marketing can be achieved.

There exist some efforts in customer segmentation in academic. For example, Jackson *et al*. [16-18] proposed to use Customer Lifetime Value (CLV) and Customer Lifetime Profit (CLP) to construct an evaluation system and take the customers with maximum net profit as high value customers. However, evaluation indicators based method is quite dependent on the experiences and domain knowledge of experts with strong subjective. Another category of methods are using clustering techniques to automatically discover potential patterns from the history data of customers. For example, Kuo *et al*. [19-20] combined fuzzy clustering and k-means clustering to find out high value customers from the customer base. However, traditional clustering is targeted to the whole set of attributes for the whole cases in one direction, that is, clusters are divided either by attributes or cases. Therefore, the clustering results are not fine enough, and the characteristics are not comprehensible since irrelevant and redundant dimensions might be involved, which contributes the low quality of clustering.

In this paper, we employ bi-clustering to improve the customer segmentation task. Specifically, based on a case study on customer segmentation for a supermarket, we proposed a bi-clustering algorithm to identify high value customers. Besides, we conduct experiments to prove the effectiveness and efficiency of bi-clustering compared to traditional clustering algorithms such as k-means.

2. Preliminary

The concept of bi-clustering (or simultaneous clustering) was first proposed by Hartigan in 1972 [10], which presented a technique for clustering cases and variables simultaneously, and the clustering results are the direct interpretation of the clusters on the data. Later on, Cheng *et al*. [21] introduced bi-clustering into gene expression pattern analysis by allowing automatic discovery of similarity based on a subset of attributes and simultaneous clustering of genes and conditions. Besides, they gave the definition of bi-cluster as follows:

Definition 1 (Bi-cluster). Suppose X is the set of genes, Y is the set of conditions, and A is the corresponding matrix of genes and conditions, where element $a_{ij} \in A$ means the i-th gene and j-th condition. Let I, J be the subset of X, Y, and the corresponding matrix for (I, J) is notated as A_{IJ}. The mean square residual of A_{IJ} is:

$$H(I,J) = \frac{1}{|I \| J|} \sum_{i \in I, j \in J} (a_{ij} - a_{i,J} - a_{I,j} + a_{I,J})^2 , \quad (1)$$

Where $a_{i,J} = \frac{1}{|J|} \sum_{j \in J} a_{ij}$, $a_{I,j} = \frac{1}{|I|} \sum_{i \in I} a_{ij}$, $a_{I,J} = \frac{1}{|I \| J|} \sum_{i \in I, j \in J} a_{ij}$ are the mean value of row, column and submatrix respectively. If sub-matrix A_{IJ} satisfies $H(I,J) \le \delta, \delta \ge 0$, A_{IJ} is termed as a δ-bi-cluster.

Therefore, the objective of bi-clustering is to discover all sub-matrices (*i.e*., bi-clusters) that satisfy the above requirement. Compared to traditional clustering, bi-clustering has two main characteristics. (1) locality: within a bi-cluster, genes have high local similarity for some conditions instead of all conditions. (2) multiplicity: one gene can be expressed by multiple conditions or none, that is one gene can belong to multiple bi-clusters or none.

Figure 1 illustrates the difference between traditional clustering and bi-clustering. Take the gene expression as an example, where rows denote genes, and columns denote

conditions. Figure 1(a) is the clustering based on genes, *i.e.*, to group the dataset by cases, and Figure 1(b) is the clustering based on conditions, *i.e.*, to group the dataset by variables. Either way is performed over the whole dataset with all the cases and variables. By contrast, bi-clustering in Figure 1(c) is dependent on both rows and columns to discover potential local patterns for specific cases and variables.

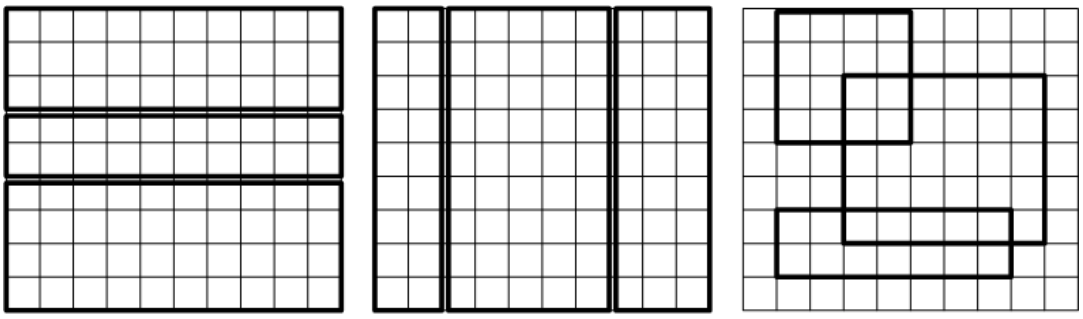

Figure 1. Traditional Clustering V.s. Bi-Clustering

3. Bi-Clustering Algorithm

Suppose the customer record is a $m \times n$ matrix, notated as A, where each row denotes a customer case, and each column denotes one attribute of customer records. Suppose A_{KL} is a sub-matrix of A, where K is the size of rows and L is the size of columns.

The objective of customer segmentation using bi-clustering is to find out the best bi-clusters with highest value and maximum support. On one hand, the best bi-clusters should have minimum mean square residual, so that the objects within one bi-cluster are consistent enough. Second, we need to find out the most significant characteristics of specific clusters, that is, to discover the set of attributes with maximum support within each bi-clusters. In this way, not only we ensure the residuals of bi-clusters are minimum, but also the attributes within each bi-cluster are most relevant.

Indicated by Shabalin *et al.* [22], matrix A can be expressed by the composition of K-layer bi-cluster and noises, where each element a_{ij} is:

$$a_{ij} = \sum_{k=1}^{K} a_k I(i \in I_k, j \in J_k) + \varepsilon_{ij}, \quad (2)$$

Where I_k, J_k are the sets of rows and columns for k-th layer bi-cluster, a_k is the value at i-th row and j-th column in the k-th layer bi-cluster, and ε_{ij} is the random noise, and subjects to standard normal distribution, *i.e.*, $\varepsilon \sim N(0,1)$. $I(i \in I_k, j \in J_k)$ is 1 if there exists element at i-th row and j-th column in the k-th layer bi-cluster; otherwise 0.

Assuming that there is no bi-clusters in matrix A, we have $a_{ij} \sim N(0,1)$, where $a_{ij} \in A$. Suppose the probability of sub-matrix A_{KL} satisfying $H(K,L) \le \delta$ is notated as $p_{KL}(\delta)$, where $H(K,L)$ is calculated from Equation (1). There are $C_m^K C_n^L$ sub-matrixes in total. Define a score function for a sub-matrix $K \times L$ as follows.

Definition 2 (Score function). Assign a score for sub-matrix $K \times L$ by:

$$Score(A_{KL}) = -\lg\left[C_m^K C_n^L p_{KL}(\delta)\right], \tag{3}$$

Which means there exists at least one submatrix whose mean square residual is smaller than δ, if there is no bi-clusters in matrix A. The larger $Score(A_{KL})$ is, the better the mean and size A_{KL} is.

On the other hand, we want the attributes (or variables) within each bi-cluster are most relevant; in other words, the support of specific attribute set within bi-clusters is maximized. We define the support as follows.

Definition 3 (Attributes support). Let the set of attributes (*i.e.*, columns) be $C \subseteq M$, where M is the whole set of attributes in A. The attributes support of C can be calculated as:

$$Sup(C) = \begin{cases} \min\limits_{j \in C} |a_{ij}|, & if\ \left[\forall j \in C, a_{ij} \neq 0\right] \wedge \\ & \left[(\max\limits_{j \in C} a_{ij} - \min\limits_{j \in C} a_{ij}) \leq \alpha \min\limits_{j \in C} |a_{ij}|\right]; \\ 0, & otherwise. \end{cases} \tag{4}$$

Where i denotes rows or cases, j denotes columns or attributes.

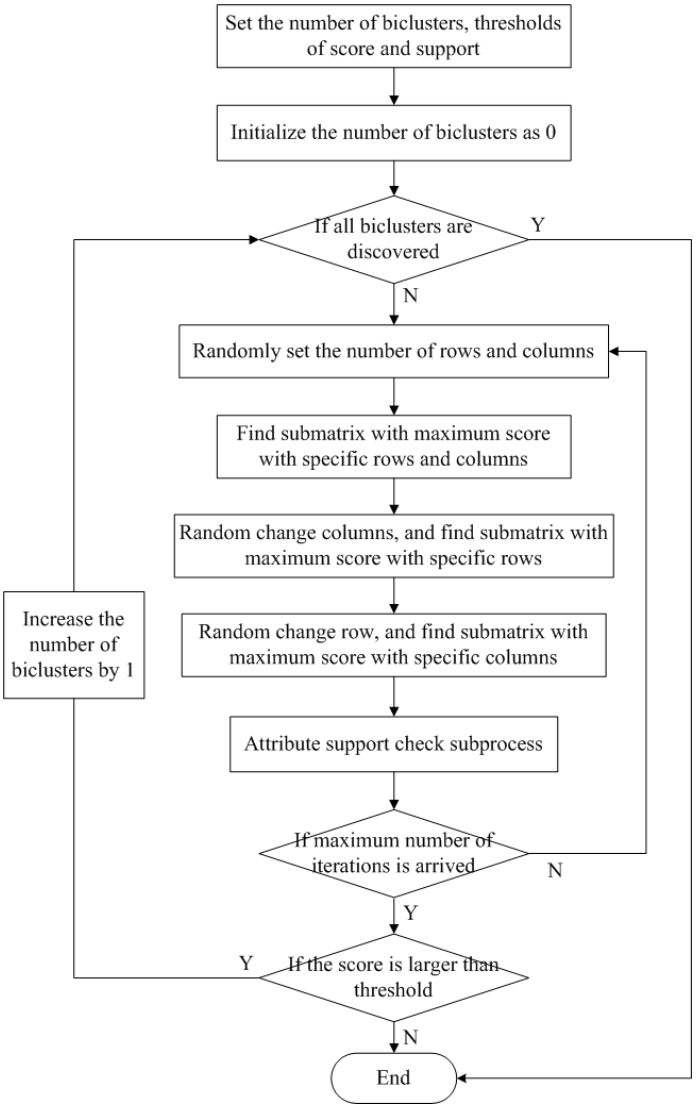

Figure 2. Flow Chart of Bi-Clustering

Attribute support $Sup(L)$ is used to discover the bi-clusters with most relevant and consistent attribute set, and also check if a bi-cluster could be merged to an existing bi-cluster. Therefore, given a matrix A composed by cases (*i.e.*, customers) and attributes, the objective of our bi-clustering algorithm is to discover the submatrix A_{KL} with maximum score $Score(A_{KL})$ and minimum support $Sup(L)$. The process of proposed bi-clustering algorithm is shown in Figure 2.

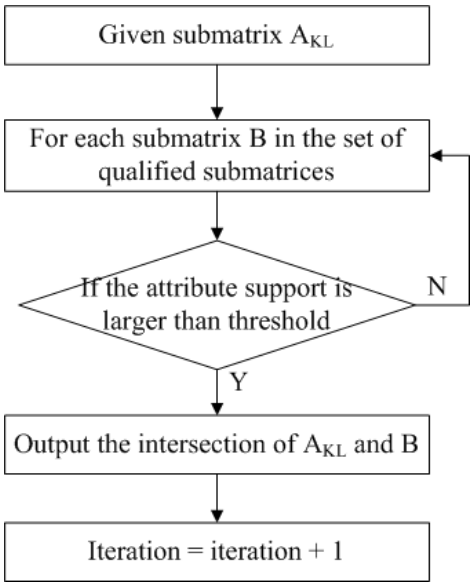

Figure 3. Flow Chart of Attribute Support Check

As shown in Figure 2, after constructing data matrix $A_{m\times n}$, the bi-clustering algorithm can be described as follows:

Step 1: input the number of bi-clusters nc, the threshold of $Score(A_{KL})$ th_{scr}, the threshold of $Sup(L)$ $th_{\sup}$, and the maximum number of iterations itr;

Step 2: randomly select K rows and L columns, where $K \in (0, m/2), L \in (0, n/2)$, and get the sub-matrix A_{KL} with maximum $Score(A_{KL})$;

Step 3: iteratively change the numbers of rows and columns in turn, and get the sub-matrix with maximum $Score(\cdot)$. Now we have the set of qualified submatrices;

Step 4: for each qualified sub-matrix given A_{KL}, check the attribute support, as shown in Figure 3. If $Sup(L)$ is larger than $th_{\sup}$, get the intersection of two submatrices, and then return to Step 2 for the next iteration;

Step 5: for discovered submatrix, check score function $Score(\cdot)$. If $Score(\cdot)$ is larger than th_{scr}, one bi-cluster is found, and then return to Step 2 to find the next bi-cluster; otherwise, the algorithm stops.

4. Experiment

The dataset we used in this experiment is FoodMart2000[1], an example database from Microsoft SQL Server 2000 Analysis Services. FoodMart is a large scale supermarket in America, Mexico and Canada. There are 10,281 customer records, and the attributes of

[1] https://msdn.microsoft.com/en-us/library/aa217032

customers include: name, sex, address, phone number, marital status, education, yearly income, number of children, card type, the amount of consumption, *etc*. Before further analysis, we first perform normalization of all fields:

$$x'_{ij} = \frac{x_{ij} - \overline{x_j}}{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \overline{x_j})^2}} . \quad (5)$$

We implement the bi-clustering algorithm using Matlab. The settings are $nc = 3$, $th_{scr} = 0$, $th_{\sup} = 60\%$, $itr = 100$. We compare the performance of proposed bi-clustering algorithm with k-means.

First, in order to measure the contributions of each cluster compared to the whole customer base, we define the value degree of each cluster as the ratio of the average amount of consumption within the cluster to the average over the whole consumption for all customers. Table 1 gives the results of value degree comparison. From Table 1 we can observe that Cluster 1 is the highest-value group, while Cluster 3 is the lowest-value group. However, using k-means clustering, the difference is not obvious, and thus k-means is not suitable for identifying high-value customers in customer segmentation problem. The reason that bi-clustering is more precise in identifying high-value customers lies in the fact that bi-clustering allows to put together the subset of attributes related to consumption ability, while k-means can only cluster over the whole set of attributes.

Table 1. Comparison of Value Degree of Clusters

Cluster No.	Bi-clustering	K-means
1	236.84	2.33
2	102.19	1.17
3	8.45	1.01

Second, we evaluate the effectiveness of clustering results. Since the number of attributes in each bi-cluster varies, we define some measurements as follows.

Definition 4 (Degree of Separation). Given the clustering results $C_1, C_2, ..., C_{nc}$, where nc is the number of bi-clusters, the degree of separation for the clustering results is calculated as:

$$DS = \sum_{i=1}^{nc} \frac{1}{|v_i|} \| c_i - c \|^2 , \quad (6)$$

Where v_i is the number of attributes in i-th bi-cluster, c_i, c are the centroids of i-th bi-cluster and the whole dataset, and $\|\cdot\|$ denotes the distance.

Definition 5 (Degree of Closeness). Given the clustering results $C_1, C_2, ..., C_{nc}$, where nc is the number of bi-clusters, the degree of closeness for the clustering results is calculated as:

$$DC = \sum_{i=1}^{nc} \max \left\{ \sum_{x \in C_i} \| x - c_i \|^2 \right\}, \quad (7)$$

Where C_i is the i-th bi-cluster, and c_i is the centroid of C_i .

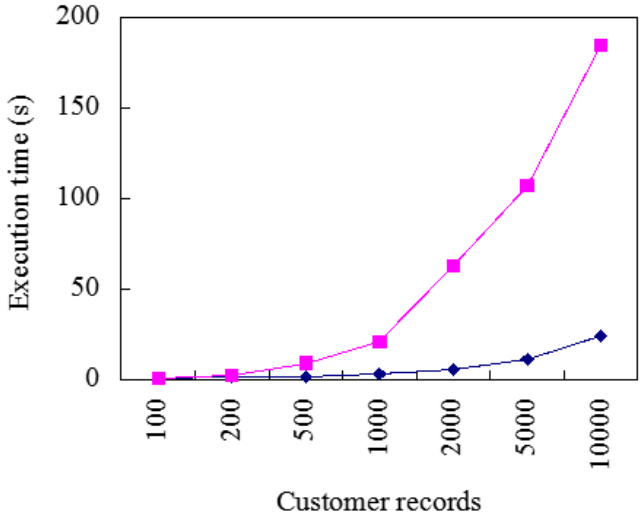

Figure 4. Execution Time Comparison of Bi-Clustering and k-Means

The larger DS, the better differentiation across clusters. The smaller DC, the more similarity within clusters. Table 2 gives the effectiveness comparison of clustering results. Calculate the measurements for each clustering result of proposed bi-clustering algorithm and k-means, we can observe that bi-clustering outperforms k-means.

Table 2. Comparison of Clustering Effectiveness

Algorithm	DS	DC
Bi-clustering	427.12	138.29
K-means	18.65	379.38

We also compare the execution time of bi-clustering and k-means in Figure 4. From the figure, we can see that with the size of dataset increasing, the execution time of bi-clustering is much less than k-means. The possible reason is that unlike k-means globally clustering over the whole dataset, bi-clustering treats the set of cases and attributes locally, which effectively reduces the problem scale. Therefore, bi-clustering is more efficient in dealing with large scale and high-dimensional dataset.

5. Conclusion

In this paper, we propose to introduce bi-clustering algorithm for customer segmentation problem, and compare the performance of proposed bi-clustering with k-means algorithm. In future works, we would like to investigate the extension of bi-clustering applications and the combination of bi-clustering with other algorithms, swarm intelligence algorithms, for example.

Acknowledgment

This work is supported by Shandong Science and Technology Development Program (2012GGX10106).

References

[1] K. Mehmed, “Data mining: concepts, models, methods, and algorithms”, John Wiley & Sons, **(2011)**.

[2] Ngai Eric W. T., L. Xiu and D. C. K. Chau, "Application of data mining techniques in customer relationship management: A literature review and classification", Expert systems with applications, vol. 36, no. 2, **(2009)**, pp. 2592-2602.
[3] Ngai E. W. T., "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature", Decision Support Systems, vol. 50, no. 3, **(2011)**, pp. 559-569.
[4] Harding J. A., M. Shahbaz and A. Kusiak, "Data mining in manufacturing: a review", Journal of Manufacturing Science and Engineering, vol. 128, no. 4, **(2006)**, pp. 969-976.
[5] L. S. Hsien, P. H. Chu and P. Y. Hsiao, "Data mining techniques and applications-A decade review from 2000 to 2011", Expert Systems with Applications, vol. 39, no. 12, **(2012)**, pp. 11303-11311.
[6] M. Fionn and P. Contreras, "Algorithms for hierarchical clustering: an overview", Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 2, no. 1, **(2012)**, pp. 86-97.
[7] K. Tapas, "An efficient k-means clustering algorithm: Analysis and implementation", Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 24, no. 7, **(2002)**, pp. 881-892.
[8] V. Juha and E. Alhoniemi, "Clustering of the self-organizing map", Neural Networks, IEEE Transactions on, vol. 11, no. 3, **(2000)**, pp. 586-600.
[9] Brown Michael P. S., "Knowledge-based analysis of microarray gene expression data by using support vector machines", Proceedings of the National Academy of Sciences, vol. 97, no. 1, **(2000)**, pp. 262-267.
[10] Hartigan J. A., "Direct clustering of a data matrix", Journal of the American statistical association, vol. 67, no. 337, **(1972)**, pp. 123-129.
[11] S. Qizheng, Y. Moreau and B. D. Moor, "Bi-clustering microarray data by Gibbs sampling", Bioinformatics 19.suppl 2, **(2003)**, pp. ii196-ii205.
[12] De Castro Pablo A. D., "Applying bi-clustering to text mining: an immune-inspired approach", Artificial Immune Systems. Springer Berlin Heidelberg, **(2007)**, pp. 83-94.
[13] De Castro Pablo A. D., "Applying bi-clustering to perform collaborative filtering", Intelligent Systems Design and Applications, 2007. ISDA 2007. Seventh International Conference on. IEEE, **(2007)**.
[14] T. Konstantinos and A. Chorianopoulos, "Data mining techniques in CRM: inside customer segmentation. John Wiley & Sons, 2011.
[15] R. F. John, "Pareto's principle-The 80-20 rule", Business Credit-New York Then Columbia Md., vol. 107, no. 7, **(2005)**, pp. 76.
[16] Jackson B. B., "Build customer relationships that last", Harvard Business Review, **(1985)**.
[17] K. S. Yeon, "Customer segmentation and strategy development based on customer lifetime value: A case study", Expert systems with applications, vol. 31, no. 1, **(2006)**, pp. 101-107.
[18] H. Hyunseok, T. Jung and E. Suh, "An LTV model and customer segmentation based on customer value: a case study on the wireless telecommunication industry", Expert systems with applications, vol. 26, no. 2, **(2004)**, pp. 181-188.
[19] Kuo R. J., L. M. Ho and C. M. Hu, "Integration of self-organizing feature map and K-means algorithm for market segmentation", Computers & Operations Research, vol. 29, no. 11, **(2002)**, pp. 1475-1493.
[20] Z. Yiyang, J. Jiao and Y. Ma, "Market segmentation for product family positioning based on fuzzy clustering", Journal of Engineering Design, vol. 18, no. 3, **(2007)**, pp. 227-241.
[21] C. Yizong and G. M. Church, "Bi-clustering of expression data", Ismb, vol. 8, **(2000)**.
[22] S. Andrey A., "Finding large average submatrices in high dimensional data", The Annals of Applied Statistics, **(2009)**, pp. 985-1012.

Authors

Chengduan Wang, he was born on March, 28, 1967 in Weifang, China. Current position, grades: Professor, Dean of school of computer engineering; University studies: Shandong Science and Technology University; Scientific interest: intelligent computing, software engineering; Publications: 5 papers, over 5 books; Experience: He received the M. Sc. degree in computer applications from Shandong Science and Technology University (2007). He is currently a professor in school of computer engineering at Weifang University, China. He has published 5 papers and over 5 books in professional fields.

