TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.7, July 2014, pp. 5529 ~ 5536
DOI: 10.11591/telkomnika.v12i7.4614

Sequence Clustering Algorithm Based on Weighed Sequential Pattern Similarity

Di Wu[1,2], Jiadong Ren*[1]
[1]College of Information Science and Engineering, Yanshan University,
Qinhuangdao 066004, China
[2]Department of Information and Electronic Engineering, Hebei University of Engineering,
HanDan 056038, China
*Corresponding author, e-mail: bestmoogoo@163.com

Abstract

Sequence clustering has become an active issue in the current scientific community. However, the clustering quality is affected heavily by selecting initial clustering centers randomly. In this paper, a new sequence similarity measurement based on weighed sequential patterns is defined. Sequence Clustering Algorithm Based on Weighed Sequential Pattern Similarity (SCWSPS) algorithm is proposed. Sequences with the largest weighted similarity are chosen as the merge objects. The last K-1 synthesis results are deleted from sequence database. Others sequences are divided into K clusters. Moreover, in each cluster, the sequence which has the largest sum of similarities with other sequences is viewed as the updated center. The experimental results and analysis show that the performance of SCWSPS is better than KSPAM and K-means in clustering quality. When the sequence scale is very large, the execution efficiency of SCWSPS is slightly worse than KSPAM and K-means.

Keywords*: data mining, sequence clustering, sequential pattern, weighted similarity*

1. Introduction

With the development of biological protein sequences analysis, user access in Web field, consumer transaction data analysis, sequence clustering has become one of the hot research problems in clustering analysis method lately [1]. The processing of sequence clustering is based on the user-defined similarity, and then the sequences of database are divided into several clusters. At the same time, the best clustering results are those similarities between sequences in each cluster as small as possible, and similarities between clusters as large as possible [2].

K-means algorithm has become one of the most popular clustering algorithms, the idea of *K*-means is very simple, and it can be realized easily. However, Euclidean distance is used in traditional *K*-means to compute the similarities between data objects [3]. When the data size is very large, it will result in too much time cost. In addition, the initial cluster centers are selected at random. If the selections are in the vicinity of the local minimum, the final generated clustering results are the local optimal solution after several times of iterative update. The expected global optimal solution can not be obtained [4].

2. Related Work

At present, many scholars have devoted to theoretical research of improved *K*-means algorithms for optimizing initial clustering centers. On the basis of the density, Sun et al. provided a genetic *K*-means based on meliorated initial center [5]. *K* objects which are belong to high density area and the most far away to each other are selected as initial centers. An enhancing *K*-means for improving initial centers was discussed by Yedla [6]. In this method, the original data points are sorted into *K* equal sets according to the sorted distances. In each set, the middle points are taken as the initial centers. *K*-means based on PSO (Particle Swarm Optimization) was proposed by Fu [7]. However, the classical PSO tends to get local optimums. Global optimized clustering partition can not be obtained. Yu et al. presented *K*-means based on improved PSO [8]. Based on particle hybridization, mutation operation, and dynamic change

Received October 3, 2013; Revised February 10, 2014; Accepted March 5, 2014

of the flight acceleration factor of particle in the state space of each dimension, improved PSO can overcome the problem of initial centers selection sensitivity.

Although the performance of the selections of initial clustering centers of above algorithms have been improved. However, the similarity measurements are not applicable to deal with sequence. In literature [9], a new similarity measurement based on bitmap operations is defined. The computation complexity can be reduced effectively. A novel sequence similarity function based on identification distance and edit distance was introduced [10]. If the identity distance between two sequences is greater than the specified threshold, it is not necessary to compute repeatedly the edit distance. The overall implementation efficiency of algorithm is improved greatly. However, the similarity measurements of these kind algorithms do not consider the weight of sequential patterns in each sequence. KSPAM (*K*-means algorithm of sequence patterns mining based on the Huffman method) [11] was presented by Yang. A highly efficient operators of 'and' and 'or' are applied to calculate similarity between sequences. Thus both of the costs of time and memory are great in KSPAM.

In this article, in order to consider the weight of sequential patterns in each sequence, a new weighted sequential pattern based similarity for measuring the pair of sequences is presented. For the purpose of obtaining the global optimal clustering results, initial clustering centers are gained according to merging the sequences with the largest weighted similarity.

The reminder of this paper is organized as follows. In section 2, we describe the related work of sequence clustering. Section 3 gives problem definitions. Section 4 concludes the SCWSPS method. Section 5 contains experimental results, and we offer our conclusions in section 6.

3. Problem Definitions

Assume that $SD=\{S_1, S_2,\cdots,S_N\}$ represents the sequence database. Wherein, N is the number of sequences in SD. Any sequence of SD is denoted as $S=a_1a_2\cdots a_n$, where a_n is the n-th item, $a_n \in L$. L indicates the set of items. Suppose that $Sup(P)$ is the number of occurrences of sequence pattern P in SD. If $Sup(P)\geq MinSup$, P is deemed to be a frequent sequence. Wherein, $MinSup$ is the user-defined minimum support threshold. The set of frequent sequential patterns in SD is represented as FSP. $FSP=\{FSP_1, FSP_2,\cdots,FSP_t\}$. FSP_t is the t-th frequent sequential pattern. t is the number of frequent sequential patterns in SD.

Assume that there is a one-to-one correspondence between weighted sequence vector $W(S_i)$ and each frequent sequential pattern in FSP. The value of each dimension of $W(S_i)$ is the weight of each frequent sequential pattern in S_i. $W(S_x)=\{W_{i1},W_{i2},\ldots,W_{ir},\ldots,W_{it}\}$. Wherein, W_{ir} denotes the weight of frequent sequential pattern FSP_t in S_i, $0\leq W_{ir}\leq 1$, $1\leq r \leq t$.

The way to measure the similarity between sequences is a critical issue for sequence clustering. On the basis of weighted sequence vector of each sequence, the weighted sequential pattern similarity $WSPSim$ (S_i,S_j) is described as follows.

Definition 1. Suppose that S_i and S_j are any two sequences in SD, the weighted sequential pattern similarity $WSPSim$ (S_i,S_j) between S_i and S_j can be designed as below:

$$WESim(S_i, S_j) = \frac{\sum_{r=1}^{t} W_{ir} W_{jr}}{\sqrt{\sum_{t=1}^{t} W_{ir}^{\ 2}} \sqrt{\sum_{t=1}^{t} W_{jr}^{\ 2}}} \quad (1)$$

$$W_{ir} = EFS_{ir}\left[\log_2\left(\frac{N}{EFD_r}\right)+1\right] \quad (2)$$

Wherein, EFS_{ir} represents the frequency of frequent sequential pattern FSP_r in S_x. If the frequencies of FSP_r in each sequence are higher, then the weight of FSP_r is larger. EFD_r indicates the number of sequences which contain FSP_r in SD. If the frequency of FSP_r in SD is higher, then the importance of FSP_r is lower. N is the number of sequences in SD.

In addition, sequence clustering criteria function E is described as follows. Where SC_j indicates the mean vector of cluster C_j.

$$E = \sum_{j=1}^{K} \sum_{S_i \in C_j} \left\| S_i - SC_j \right\|^2 \quad (3)$$

4. The Sequence Clustering Algorithm SCWSPS

Sequence Clustering Algorithm Based on weighed sequential pattern similarity named SCWSPS is discussed. In our approach, by using WSICH (Weighted-Similarity-Based Initial Centers Handling) algorithm, K initial clustering centers are gained. Moreover, by computing the sum of similarities between any sequence and other sequences in each cluster, the sequence with the largest sum of similarities is updated to the clustering center. At last, according to analyzing the clustering criteria function, the final K clusters are obtained. The flowchart of SCWSPS is shown as Figure 1.

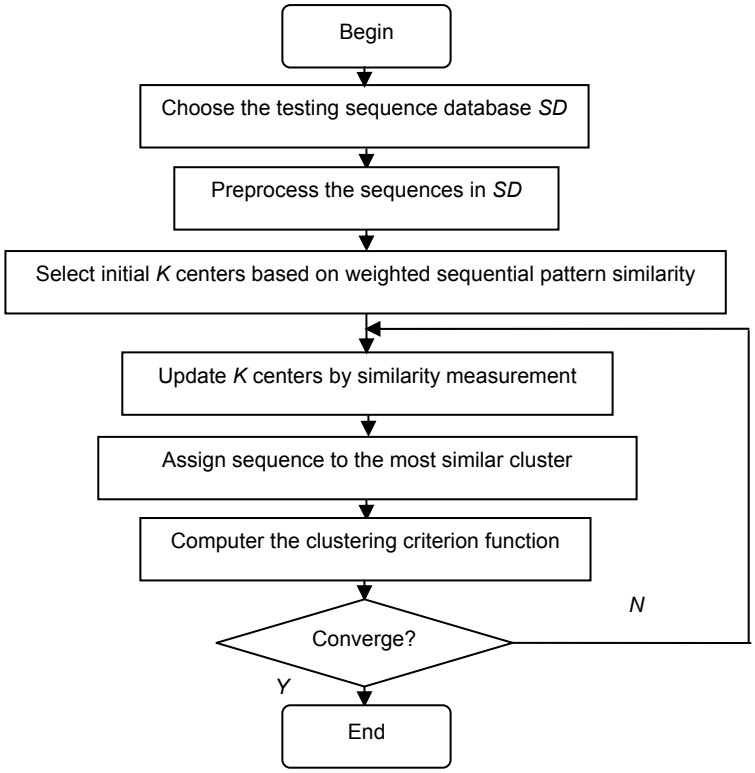

Figure 1. The Flowchart of SCWSPS

4.1. Weighted-Similarity-Based Initial Centers Handling

A novel WSICH algorithm for selecting the K initial clustering centers is given in this section. If sequences S_i and S_j has the largest similarity in current SD, then they are chosen as the merge objects. The synthesis result of S_i and S_j is denoted as S_{ij}。 S_i and S_j are deleted from SD and S_{ij} is added. Finally, the last K-1 S_{ij} are removed. And others sequences are divided into K subsets. The last generation S_{ij} or the only original sequence can be seen as the initial clustering center in each subset. The process of WSICH is explained as below:

Algorithm WSICH (SD, K, N, S_i, S_j , S_{ij})

Input: SD: Sequence database; K: The number of user-defined clusters; N: The number of sequences in SD; S_i, S_j : Any sequence in SD; S_{ij}: Synthesis result of S_i and S_j。

Output: K initial centers

Step 1: Mine the frequent sequential patterns of *SD* by Prefixspan algorithm. The frequent sequential pattern set *FSP* is get. According to the support relationships between S_i and each frequent sequential pattern, all the objects are pretreated to equal-length vectors.

Step 2: Compute the weighted similarities between any two sequences.

Step 3: In current *SD*, sequences S_i and S_j with the largest weighted similarity are chosen as the merge objects. The average of each dimensional value of the pretreated vectors of S_i and S_j is the corresponding value of synthesis result S_{ij}.

Step 4: S_i and S_j are deleted, meanwhile, S_{ij} is added to *SD*.

Step 5: Repeat Step3 and Step4, until all sequences are processed.

Step 6: On the basis of the order of merge records, the last *K*-1 S_{ij} are removed, and others sequences are divided into *K* subsets. In each subset, if there exists the last merge object S_{ij}, then S_{ij} is viewed as the initial clustering center, otherwise, the only original sequence is seen as the initial clustering center.

As to synthesis result S_{ij} represents the common frequent sequential patterns between S_i and S_j. It can be used as the representative of S_i and S_j validly. Meanwhile, in WSICH, the merge execute process is based on the simple operation of average, thus the computational complexity can be reduced greatly.

4.2. Centers Updating and Allocating

After selecting *K* initial clustering centers, the objects of *SD* are clustered to the most similar cluster. However, each initial clustering center are not the finally results. They will be updated further. In terms of the current cluster centers, the objects of *SD* are clustered to the most similar cluster. The similarities of any two sequences in *SD* are calculated. In each cluster, the sequence which has the largest sum of weighted similarities with other sequences is viewed as the new clustering center. The clustering criteria function is computed. If it is converged, then the above steps are terminated. Otherwise, continue to repeat the above steps.

On the basis of the algorithm WSICH and the procedure of centers update and allocation, sequence clustering algorithm SCWSPS based on weighted sequential pattern similarity is proposed. The specific process of SCWSPS is described as follows:

Algorithm SCWSPS (*SD*, *K*, *N*, S_i, S_j , S_{ij}, ε)

Input: *SD*: Sequence database; *K*: The number of user-defined dividing clusters; *N*: The number of sequences in *SD*; S_i, S_j : Any sequence in *SD*; S_{ij}: Synthesis result of S_i and S_j; ε*:* The clustering criteria function threshold.

Output: *K* final clusters.

Step 1: *K* initial centers are obtained by adopting WSICH algorithm.

Step 2: In terms of the current cluster centers, the objects of *SD* are clustered to the most similar cluster.

Step 3: Calculate the similarities of any two sequences in *SD*. In each cluster, the sequence which has the largest sum of similarities with other sequences is viewed as the updated clustering center.

Step 4: Compute the clustering criteria function.

Step 5: If the clustering criteria function is not converged, repeat the step2, step3 and step4. Otherwise, the algorithm is terminated. The final clustering centers are outputted.

For SCWSPS, it applies the weighted sequential pattern as an important index. The accuracy of clustering results can be greatly improved. Besides, the process of pretreatment is based on the support relationships between sequence and each frequent sequential pattern. The time cost for mining sequential patterns is very large. Thus, when handling small-scale sequence database, the performance of SCWSPS is still as good as *K*-means. However, when the scope of sequence database is very large, the execution time of SCWSPS is a little greater than *K*-means. Moreover, the selection of *K* initial clustering centers is optimized. For the clustering results of SCWSPS, the possibility of clustering results immerging in partial minimum can be decreased, further the clustering quality can be enhanced.

4.3. Case Analyzing of SCWSPS

Assume that *SD*={S_1, S_2, S_3, S_4, S_5}, wherein, S_1=abcb, S_2=aaab, S_3=acbbc, S_4=bacca, S_5=cbcaaa. The number of user-defined dividing clusters *K*=3. The minimum support *Minsup* =40%. The clustering criteria function threshold ε=0.02.

Step 1: By Prefixspan algorithm, the frequent sequential patterns of *SD* are mined. FSP={a,b,c,aa,ab,ac,ba,bb,bc,ca,cb,cc,aaa,abb,abc,acb,acc,baa,bca,cbc,cca}. In accordance with the support relationships between S_i and each frequent sequential pattern in *FSP*, all the objects are pretreated to equal-length 21-dimensional vectors.

S_1={1,1,1,0,1,1,0,1,1,0,1,0,0,1,1,1,0,0,0,0,0}; S_2={1,1,0,1,1,0,0,0,0,0,0,0,1,0,0,0,0, 0,0, 0,0}; S_3={1,1,1,0,1,1,0,1,1,0,1,1,0,1,1,1,1,1,0,0,1,0}; S_4={1,1,1,1,0,1,1,0,1,1,0,1,0,0,0,0,1,1,1,0, 1}; S_5={1,1,1,1,0,0,1,0,1,1,1,1,1,1,0,0,0,0,1,1,1,1}.

The weighted sequence vectors of each sequence are calculated.

$W(S_1)$={1,2,1.322,0,3.474,2.322,0,2.322,1.322,0,1.737,0,0, 2.322,2.322,2.322,0,0,0,0,0}. $W(S_2)$={3,1,0,5.211,5.211,0,0,0,0,0,0,0,2.322,0,0,0,0,0,0,0,0}; $W(S_3)$={1, 2, 2.644,0,3.474 ,4.644, 0, 2.322,2.644, 0,3.474,1.737,2.322, 4.644,4.644,2.322,0,0,4.644,0}; $W(S_4)$={2,1,2.644,1.737,0, 4.644,4.644,0,2.644,4.644,0,1.737,0,0,0,0,2.322,2.322,4.644,0,2.322}; $W(S_5)$={3,1,2.644,5.211, 0, 0, 6.966,0,1.322,13.932,1.737,1.737,2.322,0,0,0,0,6.966,6.966,2.322,6.966}.

According to formula (1), compute the weighted similarities between any two sequences. $WSPSim(S_1,S_2)$=0.389; $WSPSim(S_1,S_3)$=0.845; $WSPSim(S_1,S_4)$=0.228; $WSPSim(S_1, S_5)$=0.088; $WSPSim(S_2,S_3)$=0.227; $WSPSim(S_2,S_4)$=0.170; $WSPSim(S_2,S_5)$=0.240; $WSPSim$ (S_3,S_4)=0.348; $WSPSim$ (S_3,S_5)=0.137; $WSPSim(S_4,S_5)$=0.798.

The present largest similarity $WSPSim(S_1,S_3)$=0.845, S_1 and S_3 are chosen as the merge objects. $W(S_{13})$={1, 2, 1.983, 0, 3.474,3.483,0,2.322,1.983,0,2.606,0.869,0,2.322,3.483, 3.483,1.161,0,0,2.322,0}. The current $SD=\{S_{13},S_2,S_4,S_5\}$.

In the same way, $W(S_{45})$={2.5,1,2.644,3.474,0,2.322,5.805,0,1.983, 9.288,0.869,1.737, 1.161,0,0,0,1.161,4.644,5.805,1.161,4.644}. $W(S_{132})$={2, 1.5, 0.992, 2.606, 4.343,1.742,0,1.161, 0.992, 0,1.303,0.435,1.161,1.161,1.742,1.742,0.581,0,0,1.161, 0} .$W(S_{13245})$={2.25,1.25,1.818, 3.04,2.172,2.032,2.903,0.581,1.488,4.644,1.086,1.086, 1.161, 0.581,0.871,0.871, 0.871,2.322, 2.903,1.161,2.322}. According to the order of merge records, S_{13245} *and* S_{132} are deleted. Others sequences are divided into *three clusters.* The initial clustering centers are S_2, S_{13}, S_{45}.

Step 2: In cluster1, S_1 and S_3 are assigned to S_{13}. In cluster2, S_4 and S_5 are clustered to S_{45}. There is only S_2 in cluster3.

Step 3: Calculate the similarities of any two sequences in original *SD*. In each cluster, the sequence which has the largest sum of similarities with other sequences is viewed as the updated clustering center.

In cluster1, $WSPSim(S_1,S_3)$ + $WSPSim(S_1,S_{13})$= 1.782; $WSPSim(S_3,S_1)$ + $WSPSim$ (S_3,S_{13})= =1.824. Thus the updated center is S_3. In cluster2, the updated center is S_5. The present three centers are S_2, S_3, S_5.

Step 4, 5: Clustering criteria function $E=(1-0.937)^2+(1-0.979)^2+(1-0.911)^2+(1-0.976)^2=0.0129<\varepsilon=0.02$, so it is converged. The final three clustering centers are S_2, S_3, S_5.

5. Experimental Results and Analysis

In order to verify the performances of SCWSPS, *K*-means and KSPAM in literature [10], Wine recognition, yeast and segmentation-all (Complete segmentation dataset) of UCI [12] are utilized. The classic machine learning database UCI is proposed by university of California Irvine. There are 187 datasets in UCI at present. Artificial dataset is also used during comparing the clustering quality. Wherein, the relevant parameters of real and artificial datasets, part of the test sequences in artificial dataset are described as Table 1, Table 2 and Table 3, respectively.

Table 1. The Parameters of Real Datasets

Datasets	The Number of Sequences	Attribute Dimensions	The Number of Clusters
Wine Recognition	178	13	3
Yeast	1484	8	10
Segmentation-All	2310	19	7

Table 2. The Parameters of Artificial Dataset

The Number of Sequences	The Number of Clusters	The Proportions of Noises
390	3	5%

Table 3. Part of the Test Sequences in Artificial Dataset

Sequence Number	Sequence
1	acdceefb
2	debacbdfdea
3	eabcfabcdcdab
4	bdcefacbdfeacdfeacbd
5	fedcbadbacbdfddabc
…	…

Our experiments are run on the Intel Core 2 Duo 2.93 GHz CPU, 2GB main memory and Microsoft XP. All algorithms are written in MyEclipse 8.5. We compare SCWSPS with *K*-means and KSPAM in clustering quality and execution efficiency.

5.1. Clustering Quality Testing

In this section, the average correct rates of clustering results of three algorithms are tested by the formula as follows:

$$ARV(CR) = \frac{1}{q}\sum_{j=1}^{q}\frac{\frac{n_{j1}}{N_{j1}}+\frac{n_{j2}}{N_{j2}}+\cdots+\frac{n_{jK}}{N_{jK}}}{K} \quad (4)$$

Where q regards the number of tests, and the number of clusters is denoted as K. $n_{j1}/N_{j1}+n_{j2}/N_{j2}+\cdots+n_{jK}/N_{jK}$ represents the correct rate of cluster K of the j-th test. In this test, we set q=15. The average correct rates of clustering results of three algorithms in real datasets are shown as Table 4.

Table 4. The Average Correct Rates of Clustering Results in Real Datasets (ARV(CR)/%)

Datasets	SCWSPS	KSPAM	*K*-means
Wine Recognition	88.98%	82.36%	68.22%
Yeast	91.32%	84.83%	72.94%
Segmentation-All	93.37%	87.91%	75.38%

For getting the clustering quality under different minimum support, we set K=3. *Minsup*=3,6,9,12 and 15. The experimental results in artificial datasets are given as Figure 2.

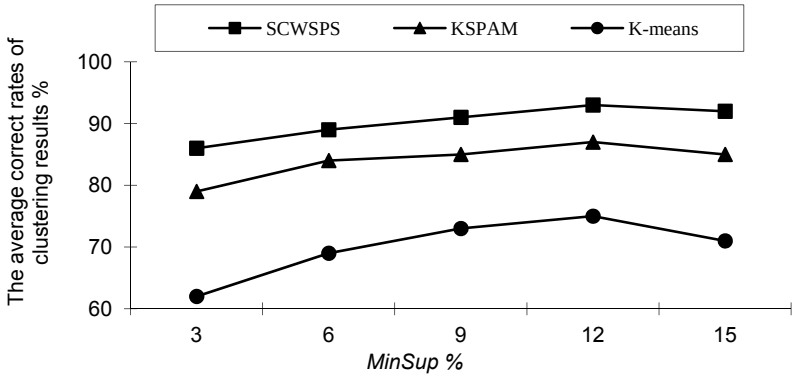

Figure 2. The Average Correct Rates Comparison of Clustering Results in Artificial Dataset

Table 3 and Figure 2 indicate that in two datasets, the average correct rates of clustering results of SCWSPS is better than the other two algorithms. For SCWSPS, the weighted of frequent sequential patterns are considered to measure the similarities of

sequences. The meanings of clustering results are explained accurately. Further the clustering quality can be greatly improved.

5.2. Execution Efficiency Analyzing

To analyze the execution efficiency of the three algorithms, UCI and artificial datasets are used. The results of the execution time in two datasets are shown as Table 5 and Table 6.

Table 5. Execution Time of Three Algorithms in Artificial Dataset (T/s)

Minimum Support *MinSup*	SCWSPS	KSPAM	*K*-means
3%	0.223	0.213	0.389
6%	0.254	0.252	0.425
9%	0.307	0.294	0.481
12%	0.354	0.342	0.568
15%	0.418	0.397	0.653

Table 6. Comparison of Execution Time in Real Datasets (T/s)

Datasets	SCWSPS	KSPAM	*K*-means
Wine Recognition	0.031	0.030	0.052
Yeast	0.388	0.367	0.499
Segmentation-All	1.903	1.691	1.109

As illustrated in Table 5 and Table 6, on the whole, the execution time of SCWSPS is little inferior than KSPAM and *K*-means in real datasets. Meanwhile, in artificial dataset, it is obvious that the time performance of SCWSPS is better than *K*-means, but little inferior than KSPAM under each *MinSup*. For SCWSPS, in the process of preconditioning sequences in SCWSPS, the time cost for mining frequent sequential patterns is very large while dealing with large scale sequence database. A highly efficient operators of 'and' and 'or' are applied in KSPAM to calculate similarities between sequences, the time cost can be reduced greatly. However, SCWSPS also has some advantages in time when the sequence scale is not large. Owing to the selection of initial centers is improved. Moreover, the merge execute process is based on the simple operation of average, thus the computational complexity is better than *K*-means in artificial dataset.

6. Conclusion

A novel sequence clustering algorithm SCWSPS based on weighed sequential pattern similarity is designed in this study. First and foremost, in accordance with the support relationship between each sequence and sequential patterns mining from *SD*, a new similarity based on the weight of sequential patterns in each sequence for measuring the sequences is defined. The clustering quality can be greatly improved. In addition, the way of selecting initial clustering centers randomly in *K*-means is changed. Initial clustering centers are gained according to merging the sequences with the largest weighted similarity. The global optimal clustering results can be obtained. Our experimental results and analysis show that the performance of SCWSPS is better than KSPAM and *K*-means in clustering quality. However, the execution efficiency of SCWSPS is a little inferior than KSPAM and *K*-means while dealing with the large scale sequence database.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.61170190), Youth Foundation of Hebei Educational Committee (No.Q2012070) and Science and Technology Research and Development Program of Handan (No:1321103077-3).

References

[1] Santhisree K, Devi DN, Bhaskar V. SEQDBSCAN: A New Sequence DBSCAN Algorithm for Clustering of Web Usage Data. *International Journal of Information Technology and Knowledge Management*. 2010; 2(2): 481-486.

[2] Zhao YZ, Liu XY, Zhao H. The K-Medoids Clustering Algorithm with Membrane Computing. *TELKOMNIKA*. 2013; 11(4): 2050-2057.
[3] Xiong YS. A Clustering Algorithm Based on Rough Set and Genetic Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(10): 5782-5788.
[4] Meng Y, Luo K, Liu JH, Jiang F. SA-Rough Sets K-means Resource Dynamic Allocation Strategy Based on Cloud Computing Environment. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(6): 1485-1489.
[5] Sun XJ, Liu XY. New Genetic K - means Clustering Algorithm Based on Meliorated Initial Center. *Computer Engineering and Applications*. 2008; 44(23): 166-168.
[6] Yedla M, Pathakota SR, Srinivasa TM. Enhancing K-means Clustering Algorithm with Improved Initial Center. *Journal of Application Research of Computers*. 2010; 1(2): 121-125.
[7] Fu T, Sun YM. PSO-based K-means Algorithm and its Application in Network Intrusion Detection System. *Computer Science*. 2011; 38(5): 54-58.
[8] Yu HT, Li X, Yao NM. Research on Optimization Method for K-means Clustering Algorithm. *Journal of Chinese Computer Systems*. 2012; 33(10): 2272-2277.
[9] Wu D, Ren JD. *K*-means Sequence Clustering Algorithm based on Top-*K* Maximal Frequent Sequence Patterns. *International Journal of Advancements in Computing Technology*. 2012; 4(20): 405-413.
[10] Ren JD, Cai BL, He HT, Hu CZ. A Method for Detecting Software Vulnerabilities Based on Clustering and Model Analyzing. *Journal of Computational Information Systems*. 2011; 7(4): 1065-1073.
[11] Yang TX, Wang ZH, Wang H, Wang LY. Research of Clustering Initial Center Selection. *Journal of Nanjing Normal University (Natural Science Edition)*. 2010; 33(4): 161-165.
[12] Zhang J, Duan F. Improved K-means algorithm with Meliorated Initial Centers. *Journal of Computer Engineering and Design*, 2013; 34(5): 1691-1694.