ICIC Express Letters
Volume **10**, Number **10**, October **2016**

ICIC International ©2016 ISSN 1881-803X
pp. 2363–2370

A WEIGHTED CLUSTERING ALGORITHM BASED ON USER'S ACCESS SEQUENCE SIMILARITY

Weina Li[1,2], Gaowei Han[1,2] and Jiadong Ren[1,2]

[1]College of Information Science and Engineering
Yanshan University
[2]The Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province
No. 438, Hebei Ave., Qinhuangdao 066004, P. R. China
wnli510@126.com

Received January 2016; accepted April 2016

Abstract. *Consider traditional clustering algorithms seldom had a research on users' visit behavior and content, and they cannot cluster users with similar visit behavior into a community easily. Behavior of user's visit keywords, visit time, and visit volume can also reflect user's interest. Therefore, setting about the log of user's access behavior, in view of visitor volume, sequences of accessing keywords and time interval between keywords, a weighted clustering algorithm based on sequence similarity which is called K-Similar is put forward. Starting from behavior of user's access to document, users are clustered, and important users are divided into a community. Then, literature needs of users in the same cluster are mined. This algorithm has been verified in users' behavior log data in CNKI, available for academic publishing. Experimental results show that K-Similar is more efficient in efficiency and accuracy. It can obtain more interesting user communities and show academic publishing demand of special users.*

Keywords: Users' access behavior, Visit volume, Time interval, Keywords sequence, Sequence similarity, Weighted clustering

1. **Introduction.** In recent years, domestic and foreign researchers have done a lot of work on user's access behavior and access content. Ding and Ma [1] proposed a new user similarity calculation algorithm called WUSC, which is based on user's access sequence to obtain users' browse path. According to network user's click flow, Zhu adopted classifier to train a behavioral model, and this model is used to predict [2]. Xu and Liu collected user's behavior log data, and then used k-means algorithm to cluster [3]. Ryu et al. first abstracted network users based on decision tree, and then clustered. This clustering algorithm can help to deal with problems of analyzing new users in the recommendation system [4].

In general, a lot of clustering algorithms have been proposed in network user's access behavior and content of all aspects. After clustering the user's access behavior, important users are got together into one class, and then acquire the demand of this kind of users to assist the publishing house to publish on demand.

Okada et al. proposed a pattern mining algorithm for numerical multidimensional event sequences, called cluster sequence mining (CSM) [5]. Brannock and Halanych gave meiofaunal community analysis by high-throughput sequencing: comparison of extraction, quality filtering, and clustering methods [6]. Lin et al. proposed a cluster based framework CluSoAF to analyze the user behaviors and then used semantic similarity to conduct resources recommendation for users [7].

Clustering with weights can truly reflect needs of users. When users are clustered, introduction of users' behavior weight can provide more reference information for users' group, so as to get more accurate clustering results. Chan et al. proposed a K-means clustering algorithm based on feature weight [8]; Wang et al. proposed a feature weighted fuzzy c-means algorithm [9]. Wang et al. [10] proposed a novel approach for key frames extraction on human action recognition from 3D video sequences. A self-adaptive weighted affinity propagation (SWAP) algorithm is proposed to extract the key frames. So a weighted sequence clustering algorithm based on user access behavior is put forward, which is based on user's needs. Consider user's access keywords sequence, time interval weight and the ratio of the number of visits, and k-means is improved, users are clustered, important users are grouped to a class, and then focus on researching needs of this kind of important user groups.

Remaining of the paper is organized as follows. In Section 2, problems are defined. Algorithms are described in Section 3. Section 4 is experiment. The paper is concluded in Section 5.

2. Definition and Architecture.

2.1. **Definition.** User's access to the literature generally includes search, browse and download, and data of these behaviors is the basis of study. Keywords which users access to are called items of users' behavior sequence, which is called a sequence of user behavior.

Definition 2.1. *Weight of users' search, browse, download amount $W(X, Y, Z)$: given user's search, browse and download volume, they are X, Y, Z. Weight of user's search, browse, download volume is defined as follows:*

$$W(X, Y, Z) = \frac{Z}{X + Y + Z} \tag{1}$$

Definition 2.2. *Time Interval between Users' Behavior. Given a user's keyword sequence $P = [a_1 \rightarrow a_2 \cdots \rightarrow a_n]$ and its corresponding timestamp list $TS(P) =< t_1, t_2, \cdots, t_n >$, time interval between two itemsets a_i and a_j $(1 \leq i < n,\ j = i + 1)$ in the sequence is defined as Formula (2).*

$$TI_{ij} = t_j - t_i. \tag{2}$$

If time interval between two keywords is larger, relationship between them is weaker. Therefore, the following time interval pruning strategy is designed to improve efficiency of the algorithm.

Time-interval-pruning. Mintiw is a user specified minimum time interval threshold. TI_{ij} is time intervals between two itemsets a_i and a_j $(1 \leq i < n,\ j = i+1)$ in user's access behavior sequence. If $TI_{ij} \geq$ mintiw, the relationship between a_i and a_j is ignored.

Definition 2.3. *Time Interval Weight between two Users' Behavior. u $(u > 0)$ is the length of a time unit, δ $(0 < \delta < 1)$ is the base of weight reducing in per unit time, and time interval weight between the item sets a_i and a_j is defined as follows.*

$$W\left(TI_{ij}\right) = \delta^{\frac{TI_{ij}}{u}} = \delta^{\frac{t_j - t_i}{u}} \tag{3}$$

Definition 2.4. *Time Interval Weight of a User's Behavior Sequence. Given a user behavior sequence $P = [a_1 \rightarrow a_2 \cdots \rightarrow a_n]$ and its corresponding timestamp list $TS(P) =< t_1, t_2, \cdots, t_n >$, time interval weight of user's behavior sequence $W(P)$ is calculated by Formula (4).*

$$W(P) = \begin{cases} \frac{1}{l} \sum\limits_{i=1}^{l-1} \sum\limits_{j=i+1}^{l} w(TI_{ij}), & (n \geq 2) \\ 1, & (n = 1) \end{cases} \tag{4}$$

Users' behavior sequence is composed of sequential basic events. Both basic events and their occurring order have an effect on the similarity of user's behavior sequences. Therefore, it is necessary to take the basic events and their order into account when defining similarity of user's behavior sequences.

Provided that $S = (s_1, s_2, \cdots, s_i, \cdots, s_n)$ is a collection of basic events of user's behavior sequence P, each item in the collection is not repeated. a_i is the number of basic event s_i $(1 \leq i \leq n)$ in user's behavior sequence P, and the identity of sequence P is $PI(P_1) = (a_1, a_2, \cdots, a_n)$.

Definition 2.5. *Weighted Global Similarity. Assume that the identifiers of user's behavior sequence P_1 and P_2 are $PI(P_1) = (a_1^1, a_2^1, \cdots, a_n^1)$ and $PI(P_1) = (a_1^2, a_2^2, \cdots, a_n^2)$. $|P_1|$ and $|P_2|$ are respectively the length of the sequence P_1 and P_2. The global similarity of P_1 and P_2 is defined as follows. If $a_i^1 > a_i^2$, $I_i^m = 1$; otherwise, $I_i^m = 0$; If $a_j^2 > a_j^1$, $I_j^t = 1$; otherwise, $I_j^t = 0$.*

$$\begin{aligned} &WGSim(P_1, P_2) \\ &= 1 - \frac{\max\left(\left[\sum\limits_{i=1}^{n} I_i^m(a_i^1, a_i^2)\right] \times W(X,Y,Z)_{P_1} \times W(P)_{P_1}, \left[\sum\limits_{j=1}^{n} I_j^t\left(a_j^2, a_j^1\right)\right] \times W(X,Y,Z)_{P_2} \times W(P)_{P_2}\right)}{\max(|P_1|, |P_2|)} \end{aligned} \tag{5}$$

Definition 2.6. *Local Similarity. Suppose that $LCS(P_1, P_2)$ is the maximal common subsequence of P_1 and P_2, and that the length of the $LCS(P_1, P_2)$ is $LLCS(P_1, P_2)$. The local similarity of P_1 and P_2 is defined as follows.*

$$LSim(P_1, P_2) = \frac{LLCS\left(P_1, P_2\right)}{\max\left(|P_1|, |P_2|\right)} \tag{6}$$

Definition 2.7. *$WSim(P_1, P_2)$. The weighted similarity of P_1 and P_2 is defined as follows, where $\theta + \varphi = 1$*

$$WSim(P_1, P_2) = \theta \times WGSim(P_1, P_2) + \varphi \times LSim(P_1, P_2) \tag{7}$$

The Strategy of Keywords Sampled to Reduce Dimension. Keyword thesaurus is obtained by word segmentation to get user's target retrieval words. Target clustering keywords are acquired by keyword whose frequency appears greater than a certain threshold. This can reduce the dimension of $[0, 1]$ sequence.

Sequence Simplified Strategy. Sequences with special few words are filtered out to obtain target clustering sequences, and clustering sequences are simplified, and sequences of small values are got rid of to reduce sparse words.

3. **K-Similar Algorithm.**

Generally speaking, behavior of download can reflect user's needs of literature more than browse and search behavior, considering the number of behavior has a certain practical significance. Time interval of user's behavior reflects continuity of user's behavior, and shows urgency of user's needs. Considering the relationship between time interval weight and behavior of keywords, K-Similar is used to cluster users who have similar behavior. Then focus on needs of user groups with similar behavior. Pseudo code of K-Similar is as follows.

3.1. **Users' behavior sequence similarity algorithm.**

User's needs can be shown through their access to keywords and their search, browse, download amount of keywords. So user's access data is extracted. Then, user's access behavior sequence is built. Finally, sequence similarity between user's access behavior sequence is calculated.

```
Algorithm: Sequence similarity algorithm
Input: User's access to keywords, user's access time, user's access amount;
Output: Distance between users (sequence similarity).
```

3.2. Weighted sequence clustering algorithm.

K-Similar is based on extended K-means. The process is that the k sequences is selected as the initial cluster center from n sequence, and weighted similarity of the remaining sequences is assigned to the most similar cluster centers. Then, clustering center of each new cluster is calculated. Repeat on this step until when result is smaller than specified threshold.

```
Algorithm: K-Similar
Input: Original sequence data list dataList, the initial cluster center centers, the new
clustering center list newCenters, the cluster helpCenterList; threshold thread when the
clustering stops;
Output: Clustering.
Begin:
 (1) read the $n$ sequence to the dataList, and use dataList[1], dataList[2], ..., dataList[$n$]
     to represent the $n$ sequence;
 (2) randomly select $k$ ($k < n$) centers, such as centers[1], centers[2], ..., centers[$k$], ...;
 (3) set the initial new clustering center list newCenters = null; cluster helpCenterList
     = null;
 (4) while (true){
 (5) calculate the weighted sequence similarity of the sequence in dataList dataList[1],
     dataList[2], ..., with $k$ clustering centers, centers[1], centers[2], ..., centers[$n$];
 (6) according to the maximum similarity of each sequence to the $k$ center point, the
     sequence was added to helpCenterList, forming helpCenterList[1], helpCenterList[2],
     ..., HelpCenterList[$k$];
 (7) the average value of each helpCenterList is calculated, and the mean sequence is
     constructed, which is added to the newCenters;
 (8) the weighted similarity between centers and newCenters is calculated by WSim;
 (9) if (WSim>thread) break;}
(10) End
```

Through the implementation of the above algorithm, important sequences with high interest are got together in a cluster.

3.3. Sequence clustering algorithm applied in academic publishing.

Behavior mode of users has been studied for a long time. Research of user's behavior data in search, reading and downloading is a new way to find users' needs. To explore needs of academic publishing has a certain practical significance to academic publishers, and scientific research workers. From researching users' behavior, clustering algorithm is used to get a specific user group. Obviously, users of a group have a very similar demand, which has certain guiding significance to academic publishing.

4. Experiment.

4.1. Data sources.

Data is collected from user's behavior log of CNKI. Users' retrieval, browsing and download records are collected. Main data elements include: the occuring time of behavior, keywords of behavior, the types of behavior operation (search, browse, download) and the number of these acts, etc. 6000 access records in some day were collected. The behavior time is recorded by the time stamp, and keywords are from user's retrieval input. Experiments are carried out in Windows 7, 8G memory space, with Java implementation.

4.2. Performance analysis of the algorithm.

Experimental performance analysis includes the following aspects: different performance of K-Similar when the experimental data, the number of cluster center K, keyword sampled parameter m, the parameter of sequence simplified sim and the data quantity n change, and performance of K-Similar algorithm with its contrast experiment S-Kmeans which also adopts keywords simplified strategy and sequence simplified strategy under different data quality, and comparison of K-Similar and S-Kmeans which also uses simplified strategy.

Other parameters in Figure 1 are $m = 100$, $k = 8$, $sim = 1$, and threaddistance $= 250$. As can be seen from Figure 1, time consumption of K-Similar is mainly on the conversion of $[0, 1]$ matrix. And with the increase of the number of sequences, the time to obtain the weight and cluster is relatively stable. This is because use of sequence simplified methods can filter out a lot of sequences with small value, which reflects good superiority of K-Similar. Other parameters in Figure 2 are $n = 6000$, $m = 100$, $k = 2$, $sim = 1$, and threaddistance $= 300$. As can be seen from Figure 2, time cost of K-Similar is mainly on the conversion of $[0, 1]$ matrix, and it has small fluctuations with the change of sequence clustering center. The time to obtain weight and execute the clustering is to fall. It can be seen that the number of cluster centers has little effect on efficiency of K-Similar algorithm.

Other parameters in Figure 3 are $n = 6000$, $m = 100$, $k = 8$, $sim = 1$, and threaddistance $= 300$. As can be seen from Figure 3, time consumption of the weighted sequence

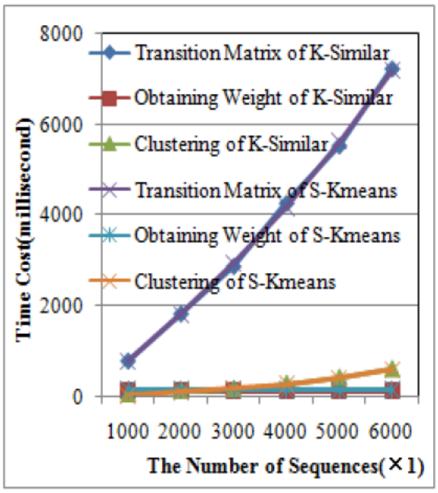


FIGURE 1. Execution efficiency when the number of sequences changes

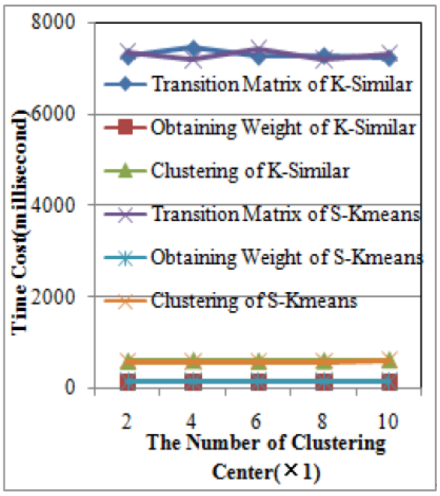


FIGURE 2. Execution efficiency when number of clustering center changes

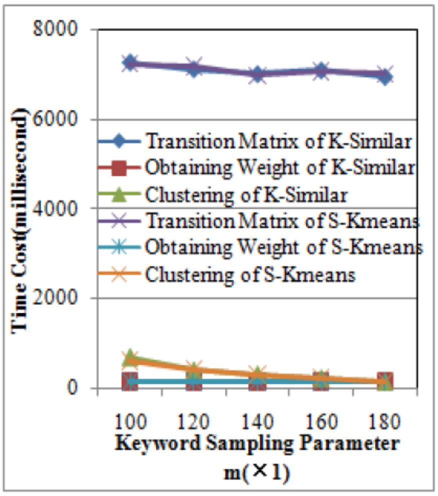


FIGURE 3. Execution efficiency when sampling parameter m changes

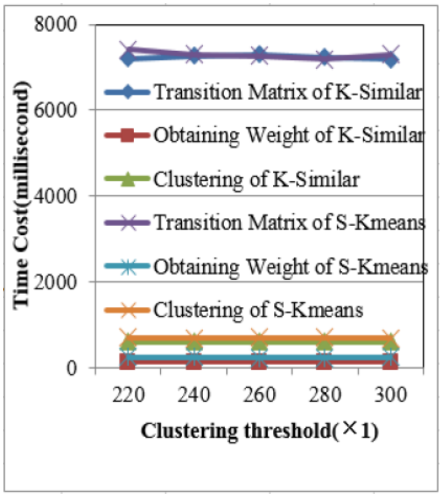


FIGURE 4. Execution efficiency when clustering threshold changes

clustering is mainly on the conversion of $[0,1]$ matrix and execution time has a downward trend with variation of sample parameters of keywords. The time to obtain weight and execution time to cluster is relatively stable. The larger the sample parameter of keywords m is, the less keywords we obtain, and the dimension of $[0,1]$ matrix is lower. And execution time is very little. Other parameters in Figure 4 are $n = 6000$, $m = 100$, $k = 8$, $sim = 1$, and threaddistance = 220. From Figure 4, we can see that the K-Similar algorithm is relatively stable when clustering threshold range ranges from 400 to 1600.

In conclusion, as can be seen from Figure 1, Figure 2, Figure 3 and Figure 4 that efficiency of K-Similar is nearly the same as that of S-Kmeans. Because keywords in S-Kmeans are also simplified, and special sequences far from clustering center are pruned to speed up the convergence rate. Performance of the algorithm is improved.

The other parameters are $n = 6000$, $m = 100$, $sim = 1$, and threaddistance = 300 in Figures 5 and 6. As can be seen from Figures 5 and 6, the number of sequences acquired in two clusters is almost the same when the number of clustering center is 10 and 12, which shows the scalability of K-Similar. However, the obtained sequence is different, and the weighted sequence clustering algorithm can cluster more sequences with bigger weight than a simple S-Kmeans algorithm. It demonstrates that the clustering accuracy of K-Similar is higher than that of S-Kmeans.

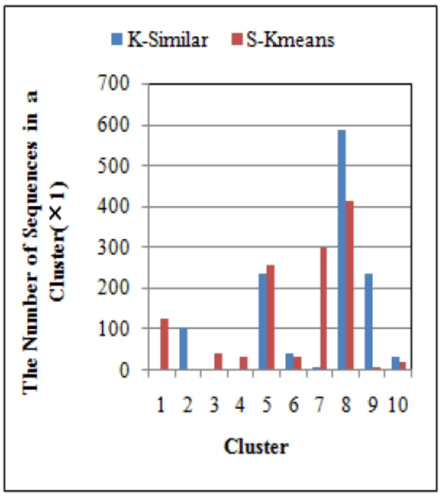


FIGURE 5. Clustering results when the clustering center is 10

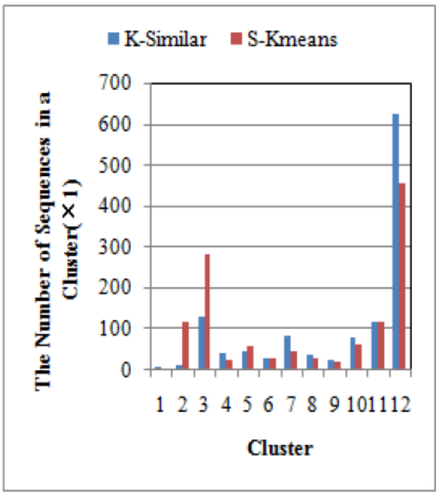
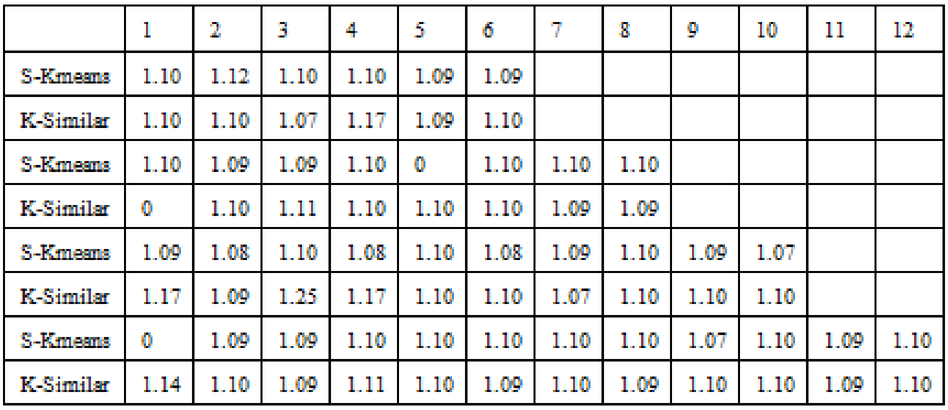


FIGURE 6. Clustering results when the clustering center is 12



	1	2	3	4	5	6	7	8	9	10	11	12
S-Kmeans	1.10	1.12	1.10	1.10	1.09	1.09						
K-Similar	1.10	1.10	1.07	1.17	1.09	1.10						
S-Kmeans	1.10	1.09	1.09	1.10	0	1.10	1.10	1.10				
K-Similar	0	1.10	1.11	1.10	1.10	1.10	1.09	1.09				
S-Kmeans	1.09	1.08	1.10	1.08	1.10	1.08	1.09	1.10	1.09	1.07		
K-Similar	1.17	1.09	1.25	1.17	1.10	1.10	1.07	1.10	1.10	1.10		
S-Kmeans	0	1.09	1.09	1.10	1.10	1.10	1.10	1.10	1.07	1.10	1.09	1.10
K-Similar	1.14	1.10	1.09	1.11	1.10	1.09	1.10	1.09	1.10	1.10	1.09	1.10

FIGURE 7. Results of K-Similar and S-Kmeans when clustering center changes

Other parameters are $n = 6000$, $m = 100$, $sim = 1$, and threaddistance = 300 in Figure 7. As can be seen from the above figures when the number of cluster center is 6, the biggest average weight of the clusters in K-Similar is 1.17. Whereas, the maximum value of the clusters in S-Kmeans is 1.12. The biggest average weight of the clusters in K-Similar is 1.11 when the number of cluster center is 8. Whereas, the maximum value of the clusters in S-Kmeans is 1.10. When the number of cluster center is 10, the biggest average weight of the clusters in K-Similar is 1.25. Whereas, the maximum value of the clusters in S-Kmeans is 1.10. The biggest average weight of the clusters in K-Similar is 1.14. Whereas, the maximum value of the clusters in S-Kmeans is 1.10 when the number of cluster center is 12, which shows that the larger the weighted average value of the weighted sequence is, the more likely the sequence is to get together than the pure S-Kmeans algorithm. It validates clustering accuracy of K-Similar is higher than that of S-Kmeans, because K-Similar uses time weight and behavior weight to cluster relatively important users into a group easily. In this way, the follow-up study on user community has a certain pertinence. The clustering accuracy degree of K-Similar is higher than that of S-Kmeans when the number of clustering center increases. Therefore, the superior scalability of K-Similar is validated.

5. **Conclusions.** Traditional clustering algorithms seldom take user's access behavior and content into consideration and it is not easy to get users who have similar access behavior together. In this paper, K-Similar algorithm is proposed. Firstly, behavior sequence of user's access to keywords is constructed. Secondly, time interval weight between items is designed, and weight of user's access amount is developed. Then, a sequence similarity algorithm is put forward. Finally, users are clustered by user's behavior sequence. Important users are clustered into a class. Special sequences far from clustering center are pruned to accelerate convergence rate of the algorithm. K-Similar uses time weight and behavior weight to cluster relatively important users into a group easily. Therefore, follow-up study on user community has a certain pertinence. Clustering relatively important users to a class plays a very important role in capturing users' demand. So the next step is to mine literature needs of the same user group, and try to further improve ability of users' demand capturing.

Acknowledgment. This work is supported by the Natural Science Foundation of P. R. China under Grant No. 61170190, No. 61472341 and the Natural Science Foundation of Hebei Province, P. R. China under Grant No. F2013203324, No. 2014203152, and No. F2015203326. The authors are also appreciated to the valuable comments and suggestions of the reviewers.

REFERENCES

[1] X. Ding and X. Ma, A web users clustering model based on users' browsing path, *IEEE Computational Intelligence and Software Engineering*, pp.1-4, 2009.

[2] T. Zhu, Clustering web users based on browsing behavior, *Proc. of the 6th International Conference on Active Media Technology*, pp.530-537, 2010.

[3] L. Xu and H. Liu, Web user clustering analysis based on K-means algorithm, *IEEE Information Networking and Automation*, pp.6-9, 2010.

[4] S. Ryu, K. Han, H. Jang and Y. I. Eom, User adaptive recommendation model by using user clustering based on decision tree, *Proc. of CIT*, pp.1346-1351, 2010.

[5] Y. Okada, K. Fukui, K. Moriyama et al., Cluster sequence mining: Causal inference with time and space proximity under uncertainty, *Advances in Knowledge Discovery and Data Mining*, pp.293-304, 2015.

[6] P. M. Brannock and K. M. Halanych, Meiofaunal community analysis by high-throughput sequencing: Comparison of extraction, quality filtering, and clustering methods, *Marine Genomics*, vol.23, pp.67-75, 2015.

[7] N. Lin, G. Chen, K. Zheng et al., CluSoAF: A cluster-based semantic oriented analyzing framework for user behaviors in mobile learning environment, *Human Centered Computing*, pp.340-351, 2015.

[8] E. Y. Chan, W. K. Ching, M. K. Ng et al., An optimization algorithm for clustering using weighted dissimilarity measures, *Pattern Recognition*, vol.37, no.5, pp.943-952, 2004.

[9] X. Wang, Y. Wang and L. Wang, Improving fuzzy c-means clustering based on feature-weight learning, *Pattern Recognition Letters*, vol.25, no.10, pp.1123-1132, 2004.

[10] Y. Wang, S. Sun and X. Ding, A self-adaptive weighted affinity propagation clustering for key frames extraction on human action recognition, *Journal of Visual Communication and Image Representation*, vol.33, pp.193-202, 2015.