A Scalable Feature Based Clustering Algorithm for Sequences with Many Distinct Items

Sangheum Hwang¹ and Dohyun Kim²

¹Department of Industrial & Information Systems Engineering, Seoul National University of Science and Technology, Seoul, Korea

²Department of Industrial and Management Engineering, Myongji University, Yongin, Korea

Abstract

Various sequence data have grown explosively in recent years. As more and more of such data become available, clustering is needed to understand the structure of sequence data. However, the existing clustering algorithms for sequence data are computationally demanding. To avoid such a problem, a feature-based clustering algorithm has been proposed. Notwithstanding that, the algorithm uses only a subset of all possible frequent sequential patterns as features, which may result in the distortion of similarities between sequences in practice, especially when dealing with sequence data with a large number of distinct items such as customer transaction data. Developed in this article is a feature-based clustering algorithm using a complete set of frequent sequential patterns as features for sequences of sets of items as well as sequences of single items which consist of many distinct items. The proposed algorithm projects sequence data into feature space whose dimension consists of a complete set of frequent sequential patterns, and then, employs K -means clustering algorithm. Experimental results show that the proposed algorithm generates more meaningful clusters than the compared algorithms regardless of the dataset and parameters such as the minimum support value of frequent sequential patterns and the number of clusters considered. Moreover, the proposed algorithm can be applied to a large sequence database since it is linearly scalable to the number of sequence data.

Keywords: Sequence data, Feature-based clustering, Frequent sequential patterns

Received: Nov. 14, 2018

Revised : Dec. 15, 2018

Accepted: Dec. 21, 2018

Correspondence to: Dohyun Kim

(ftgog@mju.ac.kr)

©The Korean Institute of Intelligent Systems

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

The amount of sequence data has grown explosively in recent years, and is expected to grow more rapidly than ever. Understanding and clustering such sequence data are invaluable for most companies in identifying different groups of customers, and thereby, develop a marketing strategy tailored to each group. In the area of biological science, grouping protein sequences that share similar structures is important because such sequences would have similar functionality.

Commonly employed algorithms for clustering sequence data are based on sequence alignment methods [1–3], which have been mainly studied in the areas of bioinformatics and computational biology to deal with such biological data as DNA and protein sequences.

Most of the existing sequence alignment methods are focused on developing effective similarity measures between sequences and efficient algorithms to calculate the measures. Then, with such computed similarity measures, any traditional hierarchical or partitional clustering algorithm can be employed. In these methods, calculation of the similarity or distance between two sequences can be reformulated as an optimal sequence alignment problem (see Section 2.1), which fits well in the framework of dynamic programming [3]. However, the quadratic computational complexity of pairwise comparison in the dynamic programming algorithm requires much computational time to obtain clustering results, which makes it impractical for most applications that require clustering of a large sequence dataset [4].

Other widely used algorithms for clustering sequence data are model-based methods. In these methods, analytical or statistical models are constructed to describe the nature of each cluster of sequences, and for that purpose, suffix trees and Markov models [5–9] are frequently used. However, these methods are not without drawbacks. They also require much computational time in general since they are iteratively implemented [10].

To overcome the drawbacks of the existing approaches, a feature-based clustering approach has been proposed. For instance, Guralnik and Karypis [4] suggested a feature-based approach which finds a set of independent frequent sequential patterns called features, and projects the sequences into a new space whose dimensions consist of those features. Then, a vector-space clustering algorithm based on K -mean is applied. To determine a set of features, they suggested a global as well as a local approach. In the global approach, two frequent sequential patterns are defined as dependent if one is sub-pattern of the other or the number of sequences which support both of them is greater than a predefined threshold. Otherwise, in a local approach, two frequent sequential patterns are regarded as dependent if they are partially overlapped in a particular sequence. With these criteria for determining the dependency between two frequent sequential patterns, they extracted a set of independent patterns from mined frequent sequential patterns to minimize the distortion of similarity.

The feature-based clustering approach has several advantages over the others. First, the high computational complexity of computing pairwise similarities (as in the dynamic programming approach) can be avoided, and therefore, a reduction in the computational time for clustering can be achieved especially for large-scale sequence databases [3]. Second, the feature-based approach deals with the global information of the whole

sequence data in the clustering process. This means that only those sequences that have frequently occurred items are clustered, while those sequences that have rarely occurred items are excluded from the clustering process even if they are much similar to each other. Consequently, it is more robust to outliers. However, the existing feature-based clustering algorithm by Guralnik and Karypis [4] uses only a subset of all possible frequent sequential patterns as features, which may result in the distortion of similarities between sequences in practice, especially when dealing with sequence data with a large number of distinct items such as customer transaction data. Moreover, it has some difficulties in dealing with sequences of sets of many distinct items. A sequence of sets of many distinct items differs from a sequence of single items in that the former consists of an ordered list of sets of many distinct items such as appeared in the sequence of customer transactions over time.

To overcome such drawbacks, a new feature-based clustering algorithm is developed for clustering of sequences with many distinct items. It is designed to deal with sequences of sets of items as well as sequences of single items. The proposed feature-based clustering algorithm regards each sequence as a binary vector in the feature space which has an orthogonal basis whose elements are the complete set of frequent sequential patterns, and employs the K -means clustering algorithm for clustering binary vectors. The proposed approach is similar to Guralnik and Karypis [4] in that it adopts the idea of projecting each sequence into a new space to treat it as a binary vector. However, in Guralnik and Karypis [4], only a subset of all possible frequent sequential patterns is used as features to be able to handle sequences with a few items (e.g., biological sequences) because these sequences may have too many possible frequent sequential patterns. However, using a subset of all possible sequential patterns may result in the distortion of similarity as can be seen in Figure 3 in Section 2.1. To deal with this problem, an additional investigation process of original sequence data to determine the independency among features is required in Guralnik and Karypis [4], while, in the proposed approach, such an additional investigation process is not required since the complete set of sequential patterns is used to minimize the distortion of similarity, and thereby, the computational time can be significantly reduced.

The rest of this paper is organized as follows: in Section 2, the proposed clustering algorithm is described in detail, and Section 3 presents the experimental and computational results. Finally, conclusions are provided in Section 4.

2. Proposed Method

The proposed algorithm for clustering sequence data consists of: 1) mining the complete set of frequent sequential patterns and simultaneously representing the sequences as binary vectors; and 2) clustering binary vectors by employing the K -means clustering algorithm. In this section, clustering procedures are discussed in detail, and then these are illustrated with an example.

2.1 Mining Frequent Sequential Patterns

Sequential pattern mining is widely used in various areas for analyzing biological sequences, customer purchase behaviors, web usage patterns, intrusion detection, etc. [11, 12]. Frequent sequential patterns obtained from the mining process can explain the nature of sequence data, and therefore, can be used as a set of features for clustering [4, 13, 14]. Several methods have been developed for efficient mining of frequent sequential patterns (e.g., see [15–18]). Agrawal and Srikant [15] may be the first who dealt with the frequent sequential pattern mining problem. Given a large database of customer transactions, where each transaction consists of customer-id, transaction time, and the items purchased by a customer, all the transactions of a specific customer can be represented as a sequence of sets of items. Given a user-defined minimum threshold, the problem of mining frequent sequential patterns is to find all subsequences that appear in the database more often than a user-defined minimum threshold.

Let $E = \{e_1, e_2, \dots, e_T\}$ be a set of T distinct items in which $e_t = t$ for $t = 1, 2, \dots, T$. Suppose that the database D of l distinct sequences is given. A sequence S_i for $i = 1, 2, \dots, l$ is an ordered list of itemsets s_{ij} for $j = 1, 2, \dots, m_i$, and is denoted by $[s_{ij}, j = 1, 2, \dots, m_i]$. An itemset s_{ij} is a set of items, and is denoted by $s_{ij} = (e_{ijk}, k = 1, 2, \dots, n_{ij}, e_{ijk} \in E)$. Without loss of generality, it is assumed that the elements in an itemset are sorted in ascending order of items. In a case of all $n_{ij} = 1$ for $j = 1, 2, \dots, m_i$, a sequence S_i , of course, can be regarded as a sequence of single items. For example, sequence S_1 in Figure 1(a) has five itemsets: $s_{11} = (1)$, $s_{12} = (2)$, $s_{13} = (3)$, $s_{14} = (4)$ and $s_{15} = (2, 2, 5, 6)$, where the elements of each itemset represent items. A sequence $S_i = [s_{i1}, s_{i2}, \dots, s_{im_i}]$ is said to support a sequence $S_{i'} = [s_{i'1}, s_{i'2}, \dots, s_{i'm_{i'}}]$ if there exist positive integers $c_1 < c_2 < \dots < c_{m_{i'}}$ such that $s_{i'1} \subseteq s_{ic_1}$, $s_{i'2} \subseteq s_{ic_2}$, $\dots$, $s_{i'm_{i'}} \subseteq s_{ic_{m_{i'}}}$, and this relationship is denoted by $S_i \supset S_{i'}$. The support value of a sequence S^* (e.g., sequential pattern) is defined as the

percentage of the sequences that support S^* . That is,

$$\text{Support}(S^*) = \frac{|S|S \in D, S \supset S^*|}{|D|} \times 100, \quad (1)$$

where $|X|$ represents the number of elements in a set X . In Figure 1, sequences $S_1 = [s_{11}, s_{12}, s_{13}, s_{14}, s_{15}] = [(1)(2)(3)(4)(2, 2, 5, 6)]$ and $S_3 = [s_{31}, s_{32}, s_{33}, s_{34}] = [(3)(7)(3, 4)(2)]$ support $p_{10} = [p_{10,1} p_{10,2}] = [(3)(2)]$ because $p_{10,1} \subseteq s_{13}$, $p_{10,2} \subseteq s_{15}$ and $p_{10,1} \subseteq s_{31}$, $p_{10,2} \subseteq s_{34}$, respectively. Therefore, the support value of p_{10} is 50%.

Given a user-defined minimum support threshold α , a set of all sequences whose support value is greater than or equal to α is called the complete set of frequent sequential patterns. The complete set of frequent sequential patterns implies that there are no constraints in the mining process except for the minimum support threshold. Obviously, the complete set includes all possible subsequences of a mined frequent sequential pattern. For example, as can be seen in Figure 1, if a sequential pattern $p_{15} = [(3)(3, 4)]$ is obtained as one of frequent sequential patterns, all subsets $p_3 = [(3)]$, $p_4 = [(4)]$, $p_{11} = [(3)(3)]$, $p_{12} = [(3, 4)]$ are also obtained.

In the present study, the PrefixSpan [16] algorithm is employed to obtain the complete set of frequent sequential patterns which is used as a set of features for clustering sequence data. A binary matrix of original sequences can be simultaneously constructed with the mining process. The PrefixSpan algorithm mines frequent sequential patterns through examining only prefixes of sequence data and projecting their corresponding postfixes into smaller projected databases. Frequent sequential patterns can be obtained by recursively building and scanning the projected databases. The process is repeated until no frequent itemset is found in the projected database. This algorithm mines all possible frequent sequential patterns (i.e., the complete set of sequential patterns) and linearly scalable. For a detailed description of the algorithm, the reader is referred to [16].

Example 2.1. Assume that the purchase records of four customers are given. In Figure 1(a), a total of 23 sequential patterns are obtained when the minimum support value is set to 50%, and they compose the complete set of frequent sequential patterns. Figure 1(b) shows a binary matrix that is constructed simultaneously with the pattern mining process.

Guralnik and Karypis [4] argued that the problem of similarity distortion is arisen because of high dependency and partial overlap between patterns. So, the original sequence data should be scanned to determine the independency or partial overlap

Figure 1. Example of constructing the binary matrix. (a) Sequence data and the complete set of frequent sequential patterns with support of 50%. (b) Constructed binary matrix.

among sequential patterns. And then, they insert additional constraints (e.g., the minimum length or the maximum length of sequential patterns) into the mining process to obtain an independent subset of the complete set of frequent sequential patterns. Then, original sequences are projected into a new space whose dimensions consist of selected sequential patterns by pruning one of dependent or partially overlapped sequential patterns. In Figure 1, sequential patterns from p_6 to p_{23} are obtained under the additional constraint that the minimum length of patterns should be greater than 1. High dependency means that if a sequential pattern (e.g., p_{14} in Figure 1) is supported by a sequence (e.g., S_3 in Figure 1), then a sub-pattern (e.g., p_3 in Figure 1) of former is also supported by the latter. Those two sequential patterns (e.g., p_{14} and p_3 in Figure 1) are called dependent. Partial overlap means that two patterns have duplicated region in a particular sequence. For example, in Figure 1, two sequential patterns p_{14} and p_{23} supported by a sequence S_3 are defined as partially overlapped patterns due to the duplicated region (Item 7) in S_3 .

However, the problem of similarity distortion is actually caused by using a subset of all possible sequential patterns (i.e., a subset of the complete set of frequent sequential patterns) as features rather than high dependency and partial overlap between patterns. In other words, the problem is mainly caused by the lack of information that sufficiently describes the whole sequential characteristic or nature. In Figure 1, Guralnik and

$$\begin{aligned}
 S_1 &= [s_{11}, s_{12}] = [(1, 2, 3)(1, 2, 4, 5)] & S_1 &= [s_{11}, s_{12}] = [(1, 2, 3)(1, 2, 4, 5)] \\
 S_2 &= [s_{21}] = [(1, 2)] & S_2 &= [s_{21}] = [(1, 2)]
 \end{aligned}$$

(a) (b)

Figure 2. Example of two possible alignments: (a) first alignment and (b) second alignment.

Karypis [4] fail to utilize the sequential patterns from p_1 to p_5 due to the additional constraint even though they have sufficient information.

Especially for the case of sequences with many distinct items, the distortion of similarity can be minimized by using the complete set of frequent sequential patterns to construct a basis of the feature space. Thus, the above unnecessary data scan process can be avoided. Moreover, there is little difference between the computational times to obtain the subset (e.g., from p_6 to p_{23} in Figure 1) and the complete set of frequent sequential patterns because most of sequential pattern mining algorithms are based on the Apriori property [19], in other words, shorter sequential patterns should be obtained first to get longer ones. Therefore, the whole computational time can be significantly reduced through eliminating unnecessary data scan process.

Before discussing the similarity distortion issue, the sequence alignment method for directly calculating the similarity between sequences is introduced. Let S_1 and S_2 be two sequences of m_1 and m_2 itemsets, respectively. First, the similarity between two itemsets s_{1i} for $i = 1, 2, \dots, m_1$ and s_{2j} for $j = 1, 2, \dots, m_2$ is defined as the ratio of the number of common items between the two itemsets to the average number of items of the two itemsets. Then, an alignment score between S_1 and S_2 can be calculated as the sum of similarities of itemsets for a possible alignment of S_1 against S_2 . The optimal alignment score is defined as the maximum alignment score between two sequences, and the similarity between the two sequences is determined as the ratio of the optimal alignment score to the average length (defined as the average number of itemsets) of the two sequences. For example, the similarity between $S_1 = [s_{11}, s_{12}] = [(1, 2, 3)(1, 2, 4, 5)]$ and $S_2 = [s_{21}] = [(1, 2)]$ can be calculated as follows. There are two possible alignments between S_1 and S_2 as depicted in Figure 2.

In the case of Figure 2(a), the alignment score is 0.8 because similarity between two itemsets s_{11} and s_{21} is 0.8. On the other hand, in the case of Figure 2(b), the alignment score is 0.67. Therefore, the optimal alignment score is 0.8, and then the similarity between S_1 and S_2 can be calculated as the optimal

Table 1. Comparison of similarity matrices using matrix norms

	Optimal alignment score (B)	Complete set (C)	Difference (B-C)	Subset length= 2 (D)	Difference (B-D)	Subset length= 3 (E)	Difference (B-E)
1-Norm	2.780	2.730	0.050	2.490	0.290	2.200	0.580
2-Norm	2.649	2.484	0.166	2.242	0.408	1.999	0.650
Infinity norm	2.780	2.730	0.050	2.490	0.290	2.200	0.580
Frobenius norm	2.804	2.728	0.076	2.606	0.198	2.614	0.190

alignment score divided by the average number of itemsets of two sequences, i.e., 0.53.

In Figure 3, the distortion of similarity issue is illustrated with the previous example. The elements of matrices represent the similarity between sequences. Figure 3(b) shows the similarity matrix obtained by the sequence alignment method and provides the optimal sequence similarity score based on the dynamic programming. Figure 3(c) is the proposed method based on the complete set of frequent sequential patterns. And Figures 3(d), and 3(e) show similarity matrices of projected sequences (e.g., see Figure 1(b)) computed by ‘cosine’ similarity under the different constraints on the minimum length of sequential patterns. The cosine similarity can be defined as follows. Let X and Y are vectors of n dimensions. The cosine similarity between X and Y is

$$\frac{\langle X \cdot Y \rangle}{\sqrt{\langle X \cdot X \rangle} \sqrt{\langle Y \cdot Y \rangle}}, \quad (2)$$

where $\langle X \cdot Y \rangle$ represents an inner product between X and Y . For example, similarity between projected sequence data S_1 and S_3 in Figure 1(b) can be calculated as follows:

$$\text{sim}(S_1, S_3) = \frac{\langle S_1 \cdot S_3 \rangle}{\sqrt{\langle S_1 \cdot S_1 \rangle} \sqrt{\langle S_3 \cdot S_3 \rangle}} = \frac{7}{\sqrt{12} \sqrt{18}} = 0.48. \quad (3)$$

To measure the distortion of similarity in Figure 3, various matrix norms including 1-norm, 2-norm, infinity norm and Frobenius norm were used. Matrix norms are generally used to quantify the difference between two matrices. For a detailed description of the matrix norms, the reader is referred to Kim et al. [20, 21]. From the comparisons using the matrix norms (Table 1), we can find using a complete set of frequent sequential patterns as features (C in Table 1) has the smallest difference from the optimal alignment score based on the dynamic programming (B in Table 1) regardless of the choice of norm. That is, the proposed method causes less distortion of

$S_1 = [(1)(2)(3)(4)(2,2,5,6)]$	S_1	1	0.42	0.46	0.60
$S_2 = [(3)(7)(3,4)]$	S_2	0.42	1	0.86	0.50
$S_3 = [(3)(7)(3,4)(2)]$	S_3	0.46	0.86	1	0.44
$S_4 = [(1)(2)(3)(3)(7)]$	S_4	0.60	0.50	0.44	1
(a)	(b)				

S_1	S_2	S_3	S_4	S_1	S_2	S_3	S_4	S_1	S_2	S_3	S_4
1	0.23	0.48	0.64	1	0.11	0.38	0.58	1	0	0.29	0.71
0.23	1	0.88	0.34	0.11	1	0.89	0.25	0	1	0.91	0
0.48	0.88	1	0.37	0.38	0.89	1	0.22	0.29	0.91	1	0
0.64	0.34	0.37	1	0.58	0.25	0.22	1	0.71	0	0	1
(c)	(d)	(e)									

Figure 3. Similarity distortion problem with constrained patterns. (a) Sequence data. (b) Similarity based on the optimal alignment score. (c) Similarity based on a complete set (constraints: support= 50%). (d) Similarity based on a subset of the complete set (constraints: support= 50% min length= 2). (e) Similarity based on a subset of the complete set (constraints: support= 50% min length= 3).

similarity than existing feature-based method using a subset of frequent sequential patterns (D and E in Table 1).

2.2 K-Means Clustering

The binary matrix obtained from the mining process is used for clustering the original sequences. By treating original sequences as binary vectors, the computation time for calculating similarity between sequences can be significantly reduced compared to the sequence alignment method [3].

Rows of the binary matrix represent the original sequences in the feature space. The proposed method employs the K -means clustering algorithm with the cosine distance, commonly used for document clustering [22]. Detailed procedures are as follows:

1. Randomly select K rows of the binary matrix as initial cluster centroids.

2. Assign each point to its closest centroid by computing the cosine distance between the point and each cluster centroid.
3. Compute a new cluster centroid of each cluster, which is the mean vector of the points in that cluster after normalizing the length of each points to the unit Euclidean length.
4. Repeat Steps 2 and 3 until cluster centroids converge.

The time complexity of the K -mean clustering algorithm is $O(lKI)$, where l is the number of cases (i.e., sequences in the present problem), K is the number of clusters, and I is the number of iterations of the algorithm to converge. Since l is usually much larger than both K and I , the time complexity becomes near linear to l [23].

3. Experimental Results

In order to verify the effectiveness and accuracy of the proposed method (i.e., the feature-based approach plus K -means clustering), computational experiments were conducted using a real dataset and a synthetic dataset. There are some difficulties in directly comparing the proposed method with Guralnik and Karypis [4] because there exist some ambiguities in choosing parameters such as the minimum and the maximum length of sequential patterns or a predefined threshold in the global approach. Therefore, the proposed method was compared to: 1) the sequence alignment method combined with K -medoid clustering; 2) the sequence alignment method combined with hierarchical clustering; and 3) the sequence alignment method combined with the density peaks clustering.

The K -medoids clustering algorithm [24] is similar to K -mean clustering algorithm except for the step of calculating cluster centers. The K -medoids clustering algorithm regards the cluster center of each cluster as one of the existing points that is located most centric, while in the K -mean clustering algorithm, the cluster center of each cluster is computed as the mean vector of those points in that cluster. It is impossible to compute the center of each cluster in sequence data, and therefore, the K -medoids clustering algorithm in which cluster centers need not be computed is compared with the proposed method.

The hierarchical clustering algorithms can be classified into three categories according to how the distance between clusters is defined. They include single-linkage, average-linkage, and

complete-linkage methods [25]. Sometimes, the single-linkage method has a tendency to group a large number of sequences into a single large cluster. This phenomenon is called the “chain effect”. The single-linkage method was not considered since both datasets show a high “chain effect” when the method was used. Accordingly, the average-linkage and complete-linkage methods are compared with the proposed method.

The density peak clustering algorithm assumes that the cluster centers have higher densities than their neighbors and they are at a relatively large distance from points with higher densities [26]. This approach has an advantage that the cluster centers can be intuitively selected and the clusters are recognized easily regardless of their shape and of the dimensionality of data. Moreover, this algorithm can be conducted based only on the similarity matrix. For this comparison, the cluster centers are selected using γ computed by ρ (density) $\times \delta$ (minimum distance between the sequence and any other sequence with higher density), ρ and δ are computed on the basis of the similarity between sequences. For a detailed description of the density clustering method, the reader is referred to Rodriguez and Laio [26].

3.1 Measures of Cluster Quality

There are two types of measures for validating clustering results. The external quality measure compares the discovered clusters to a priori clusters. On the other hand, the internal quality measure evaluates if the discovered clusters are inherently appropriate for the data without reference to external knowledge. Assuming no a priori knowledge on the clusters of the two experimental datasets, we adopt the internal quality measure in the present study.

The internal measure of cluster quality used in the present study is a weighted average similarity where the similarity between sequences is based on the optimal alignment score discussed in Section 2.1.

Let $C = \{C_1, C_2, \dots, C_K\}$ be a set of clusters. The weighted average similarity is calculated as follows. A higher WAS value represents a better clustering solution.

1) Compute the average similarity (AS) for each cluster C_k for $1 \leq k \leq K$.

$$AS(C_k) = \frac{\sum sim(S_i, S_{i'})}{|C_k|(|C_k| - 1)} \quad (4)$$

for all $S_i, S_{i'} \in C_k$ and $i < i'$,

where $sim(S_i, S_{i'})$ is the similarity between sequences S_i and

$S_{i'}$.

2) Calculate the weighted average similarity (WAS) of clusters C as follows.

$$WAS(C) = \sum_{k=1}^K |C_k| \cdot AS(C_k). \quad (5)$$

3.2 Retail Market Dataset

The retail market dataset used in the experiment was obtained from an e-commerce company. The dataset contains 57,686 transactions on 305 items by 5,001 customers during a six-month period in a certain year. In addition, 5,001 customers who made 3 or more transactions were considered. As a pre-processing step, we generated a sequence which is a list of transactions in ascending order of transaction time for each customer. Subsequently, 5,001 sequences of sets of items were generated, and therefore, this data can be categorized as “sequences of sets of items with many distinct items”.

To analyze the sensitivity of clustering results to the minimum support threshold, two different support threshold values, 1% and 3%, were considered. In this dataset, we found 657 and 110 frequent sequential patterns with the minimum support of 1% and 3%, respectively. Only the sequences that support at least one of the mined frequent sequential patterns were clustered to compare the five clustering algorithms on the same condition. Those sequences that support none of the patterns are treated as outliers. As a result, 4 original sequences were excluded with the minimum support of 1%, and 96 sequences were excluded with 3%. Then, the WASs were calculated when the number of clusters are 10, 20, and 30, respectively.

Figures 4(a) and 4(b) show the WASs for the five clustering algorithms. Experimental results show that the proposed clustering method performs better than the sequence alignment methods combined with other traditional and recent clustering algorithms regardless of the support value and the number of clusters. It is interesting to note that the WAS of the proposed feature-based clustering combined with K -means clustering is much higher even than that of the recent clustering algorithm based on optimal alignment score. This implies that clustering task is readily done in a vector space spanned by binary vectors obtained from the complete set of frequent sequential patterns and the proposed feature-based approach better reflects the nature of the global sequence by using the complete set of frequent sequential patterns.

Figure 4. Experimental results for the retail market dataset with the support of 1% (a) and 3% (b).

3.3 Synthetic Dataset

The synthetic dataset T10I4D100K.dat [27] generated by IBM Almaden Quest research group was used in this study. This dataset contains 100,000 transactional data with 1,000 items. We randomly chose 5,000 transactional data to reduce the time for calculating the weighted average similarity. Actually, this dataset is not a sequential one, but just a transactional one. That is, the order of items is not important. Without loss of generality, we assumed that each sequence represents a sequence of items, and treated the dataset as “sequences of items with many distinct items”.

In a similar manner to the previous experiment, we considered two support threshold values of 0.5% and 1%, and found 1,406 and 419 frequent sequential patterns, respectively. Every sequence supported at least one of the patterns with the support of 0.5%, and 4 sequences do not support any sequential patterns with the support of 1%. As mentioned earlier, sequences that support none of the sequential patterns are treated as outliers,

Figure 5. Experimental results for the synthetic dataset with the support of 0.5% (a) and 1% (b).

and thus excluded.

Figures 5(a) and 5(b) show the weighted average similarities for the five clustering algorithms. Experimental results show that the proposed clustering method performs better than the sequence alignment methods combined with other clustering algorithms regardless of the support value and the number of clusters. Compared to the previous experiment on sequences of a set of items, these results show that the quality of clusters which are obtained by the proposed method is much better than other approaches, implying that it can be used as a useful alternative when clustering sequences with many distinct items.

4. Conclusion

In this paper, a feature-based clustering algorithm for sequences with many distinct items is developed. Given two parameters, a minimum support threshold and the number of clusters, the proposed method finds meaningful clusters of sequence. It uses a complete set of frequent sequential patterns as features

for clustering and employs K -means clustering with cosine distance to obtain meaningful clusters. Unlike the existing approaches which have difficulty in dealing with sequences of sets of items, the proposed method can deal with sequences of sets of items as well as sequences of items if there exist many distinct items. Moreover, it can be applied to large scale databases since it is linearly scalable.

To verify the effectiveness of the proposed method, it is compared to other traditional and recent clustering algorithms in terms of the weighted average similarity. Two datasets were used in the experiment. One is a retail market dataset which consists of sequences of sets of items with many distinct items, and the other is a synthetic dataset of sequences of single items with many distinct items. Experimental results show that a projection of sequences into a binary vector space helps to find better clusters, and thereby the proposed method outperforms the sequence alignment based clustering approaches regardless of the support threshold value and the number of clusters considered.

The above encouraging results require the proper value of minimum support threshold for sequential patterns. Therefore, it may be a fruitful area of future research to develop an efficient method for determining a proper support value. In addition, to reduce the computational time, a direct clustering method through merging two processes mining frequent sequential patterns and clustering projected binary vectors can be developed in further study.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgement

This study was supported by the Research fund for a new professor by the Seoul National University of Science and Technology (SeoulTech).

References

- [1] S. K. Noh, Y. M. Kim, D. K. Kim, and B. N. Noh, "An efficient similarity measure for clustering of categorical sequences," in *AI 2006: Advances in Artificial Intelligence*. Heidelberg: Springer, 2006, pp. 372-382. https://doi.org/10.1007/11941439_41

- [2] S. J. Oh and J. Y. Kim, "A hierarchical clustering algorithm for categorical sequence data," *Information Processing Letters*, vol. 91, no. 3, pp. 135-140, 2004. <https://doi.org/10.1016/j.ipl.2004.04.002>
- [3] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645-678, 2005. <https://doi.org/10.1109/TNN.2005.845141>
- [4] V. Guralnik and G. Karypis, "A scalable algorithm for clustering sequential data," in *Proceedings of IEEE International Conference on Data Mining*, San Jose, CA, 2001, pp. 179-186. <https://doi.org/10.1109/ICDM.2001.989516>
- [5] M. Bicego, V. Murino, and M. A. T. Figueiredo, "Similarity-based clustering of sequences using hidden Markov models," in *Machine Learning and Data Mining in Pattern Recognition*. Heidelberg: Springer, 2003, pp. 86-95. https://doi.org/10.1007/3-540-45065-3_8
- [6] L. de Angelis and J. G. Diasb, "Mining categorical sequences from data using a hybrid clustering method," *European Journal of Operational Research*, vol. 234, no. 3, pp. 720-730, 2014. <https://doi.org/10.1016/j.ejor.2013.11.002>
- [7] P. Smyth, "Clustering sequences with hidden Markov models," *Advances in Neural Information Processing Systems*, vol. 9, pp. 648-654, 1997.
- [8] T. Xiong, S. Wang, Q. Jiang, and J. Z. Huang, "A novel variable-order Markov model for clustering categorical sequences," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2339-2353, 2004. <https://doi.org/10.1109/TKDE.2013.104>
- [9] J. Yang and W. Wang, "CLUSEQ: efficient and effective sequence clustering," in *Proceedings of the 19th International Conference on Data Engineering*, Bangalore, India, 2003, pp. 101-112. <https://doi.org/10.1109/ICDE.2003.1260785>
- [10] A. J. Enright and C. A. Ouzounis, "GeneRAGE: a robust algorithm for sequence clustering and domain detection," *Bioinformatics*, vol. 16, no. 5, pp. 451-457, 2000. <https://doi.org/10.1093/bioinformatics/16.5.451>
- [11] M. Kim, "Simultaneous learning of sentence clustering and class prediction for improved document classification," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 17, no. 1, pp. 35-42, 2007. <https://doi.org/10.5391/IJFIS.2017.17.1.35>
- [12] S. Laxman and P. S. Sastry, "A survey of temporal data mining," *Sadhana*, vol. 31, no. 2, pp. 173-198, 2006. <https://doi.org/10.1007/BF02719780>
- [13] T. Morzy, M. Wojciechowski, and M. Zakrzewicz, "Scalable hierarchical clustering method for sequences of categorical values," in *Advances in Knowledge Discovery and Data Mining*. Heidelberg: Springer, 2001, pp. 282-293. https://doi.org/10.1007/3-540-45357-1_31
- [14] Y. Yang and B. Padmanabhan, "Segmenting customer transactions using a pattern-based clustering approach," in *Proceedings of the 3rd IEEE International Conference on Data Mining*, Melbourne, FL, 2003, pp. 411-418. <https://doi.org/10.1109/ICDM.2003.1250947>
- [15] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the 11th International Conference on Data Engineering*, Taipei, Taiwan, 1995, pp. 3-14. <https://doi.org/10.1109/ICDE.1995.380415>
- [16] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth," in *Proceedings of the 17th International Conference on Data Engineering*, Heidelberg, Germany, 2001, pp. 215-224. <https://doi.org/10.1109/ICDE.2001.914830>
- [17] M. J. Zaki, "SPADE: an efficient algorithm for mining frequent sequences," *Machine Learning*, vol. 42, no. 1-2, pp. 31-60, 2001. <https://doi.org/10.1023/A:1007652502315>
- [18] Z. Zhang, J. Huang, and Y. Wei, "FI-FG: frequent item sets mining from datasets with high number of transactions by granular computing and fuzzy set theory," *Mathematical Problems in Engineering*, vol. 2005, article ID. 623240, 2005. <http://doi.org/10.1155/2015/623240>
- [19] Q. Zhao and S. S. Bhowmick, "Sequential pattern mining: a survey," Nanyang Technological University, Singapore, *Technical Report No. 2003118*, 2003.
- [20] D. Kim, B. Lee, H. J. Lee, S. P. Lee, Y. H. Moon, and M. K. Jeong, "Automated detection of influential patents using singular values," *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 4, pp. 723-733, 2002. <https://doi.org/10.1109/TASE.2012.2210214>

- [21] G. A. Seber, *A Matrix Handbook for Statisticians*. Hoboken, NJ: John Wiley & Sons, 2008.
- [22] Y. Zhao and G. Karypis, "Criterion functions for document clustering: experiments and analysis," Department of Computer Science, University of Minnesota, MN, *Technical Report No. 01-040*, 2001.
- [23] M. Kantardzic, *Data Mining: Concepts, Models, Methods and Algorithms*. Hoboken, NJ: John Wiley & Sons, 2002.
- [24] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer, 2001.
- [25] G. Dong and J. Pei, *Sequence Data Mining*. New York, NY: Springer, 2007.
- [26] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492-1496, 2014. <http://doi.org/10.1126/science.1242072>
- [27] Frequent Itemset Mining Dataset Repository (FIMI Repository), <http://fimi.cs.helsinki.fi/data/>

Sangheum Hwang received the Ph.D. degree in industrial and systems engineering from KAIST, Korea, in 2012. Currently, he is an Assistant Professor with the Department of Industrial and Information Systems Engineering, Seoul National University of Science and Technology. His research interests are in the areas of statistical learning methods, kernel machines and deep learning.
E-mail: shwang@seoultech.ac.kr

Dohyun Kim received the M.S. and Ph.D. degrees in industrial engineering from KAIST, Korea, in 2002 and 2007, respectively. Currently, he is an Associate Professor with the Department of Industrial and Management Engineering, Myongji University. His research interests include statistical data mining, deep learning and graph data analysis.
E-mail: ftgog@mju.ac.kr