

Clustering Categorical Sequences with Variable-Length Tuples Representation

Liang Yuan¹, Zhiling Hong^{2(✉)}, Lifei Chen³, and Qiang Cai⁴

¹ Network Operation Maintenance Center,
University of Electronic Science and Technology of China,
Chengdu 611731, China

² Software School, Xiamen University, Xiamen 361005, China
hongzl@xmu.edu.cn

³ School of Mathematics and Computer Science,
Fujian Normal University, Fuzhou 350117, Fujian, China

⁴ Technique Department, Xiamen Customs, Xiamen 361000, China

Abstract. Clustering categorical sequences is currently a difficult problem due to the lack of an efficient representation model for sequences. Unlike the existing models, which mainly focus on the fixed-length tuples representation, in this paper, a new representation model on the variable-length tuples is proposed. The variable-length tuples are obtained using a pruning method applied to delete the redundant tuples from the suffix tree, which is created for the fixed-length tuples with a large memory-length of sequences, in terms of the entropy-based measure evaluating the redundancy of tuples. A partitioning algorithm for clustering categorical sequences is then defined based on the normalized representation using tuples collected from the pruned tree. Experimental studies on six real-world sequence sets show the effectiveness and suitability of the proposed method for subsequence-based clustering.

Keywords: Sequence clustering · Representation model · Variable-length tuples · Pruning method · Entropy-based measure

1 Introduction

Data clustering has a wide range of applications and has been one of the essential methods used in knowledge systems. In the past decades, it was studied extensively in the statistics, machine learning and data mining communities, and a number of clustering algorithms have been proposed [1, 2]. However, most of them are principally designed for attribute-value data, say, vector data. Currently, categorical sequences, such as speech sequences in natural language processing, are widely used in real-world applications. Clustering such sequences is a difficult problem due to the fact that the chronological order of symbols (categories) that compose the sequences is very important for clustering tasks; this remains a major obstacle in applying the traditional clustering algorithms [3–5].

A widely accepted solution to the problem is using a tuples-based representation for sequences, which effectively equates to project each sequence onto the new pattern space spanned by a set of tuples [4,6]. Roughly speaking, the tuple of sequences is one kind of short subsequences; thus, the locally chronological order of symbols can be preserved to some extent. Such a representation model is somewhat similar to the Vector Space Model (VSM) for representing documents in text mining [7], where each term in the documents is considered as a dimension and each document is typically represented as a vector of the term frequencies. With the tuples-based representation, sequences are viewed as “documents” with the tuples representing their “terms”, and can thus be clustered like the common vector data.

Obviously, the ability of the representation model to capture the structural features hidden within sequences depends on the tuples chosen for the model. The common method is the n -tuples (alternatively known as n -grams) approach [4,8,9], which is the set of all possible tuples (grams) with their length fixed at n . As choosing an appropriate tuple length is currently a difficult problem, generally, one tends to use a large n , because small n likely breaks long sequence patterns into small segments [10]. However, a large n would result in a huge number of tuples which is exponential in the length. More importantly, with a fixed length, all tuples of length n are collected without distinguishing between significant and non-significant tuples [5], which challenges the traditional clustering algorithms by the existence of many noisy features (tuples) or redundant features (tuples) that do not contribute to clustering.

The popular approach adapting the algorithms to the high-dimensional data is to eliminate these features by combining feature selection techniques, for example, by removing those tuples whose frequency are less than the user-defined threshold [10]. Clearly, such a threshold is difficult to determine. Another approach is to perform subspace clustering on the high-dimensional data: examples include entropy-weighting K -means (EWKM) [11], model-based projective clustering (MPC) [12], etc. Note that such feature-weighting-based algorithms are designed on the assumption that each of the dimensions (here, the tuples) spanning the new pattern space is independent of the others, which hardly holds in the n -tuples representation for sequences: tuples that share the same preceding subsequence may be highly correlated with each other.

In this paper, a new method is proposed to produce the variable-length tuples, with a large number of redundant tuples removed. We propose a pruning method for the purpose, by organizing the original n -tuples into a suffix tree, on which those leaves corresponding to the redundant tuples are iteratively deleted in terms of the information gain provided to their parent (i.e., the preceding tuples). The remaining tuples in the pruned tree are then collected to create a normalized representation model, with which a partitioning algorithm is defined for the clustering task. We conducted a series of experiments on real-world categorical sequences. The results show that the proposed method significantly outperforms other mainstream methods.

The remainder of this paper is organized as follows. Section 2 presents some preliminaries and related work. Section 3 describes the new representation model and the clustering algorithm. Experimental results are presented in Sect. 4. Finally, Sect. 5 gives our conclusion and discusses directions for future work.

2 Preliminaries and Related Work

A categorical sequence is a linear chain made up of symbols, containing some structural features. Figure 1 gives an example, where two sequences denoted by s_1 and s_2 are shown. Both s_1 and s_2 are made up of 3 symbols “A”, “B” and “C”, but they have different lengths, saying, 14 and 12, respectively. Clustering such sequences is a challenging problem due to the difficulties in defining a meaningful distance measure for sequences [1, 4]. The existing measures fall in two groups: alignment-based and alignment-free measures [6, 13]. In the first group, the distance is computed by an alignment algorithm, such as the well-known edit distance and its approximate algorithms [14]. Generally, they have a high time complexity. The alignment-free measures in the second group calculate the distance between sequences based on statistical models [3, 10], information theory [9] or subsequences [5], without identifying the similar regions of sequences; thus, they are computationally efficient.

s_1 : ABAABBACACBACB
 s_2 : ACBABAABBACB

Fig. 1. An example of categorial sequences made up of 3 symbols “A”, “B” and “C”.

To define an alignment-free distance measure for sequences, the tuple-based representation has been widely used due to its simplicity [4, 6]. Using the model, each sequence can be transformed into a vector of tuple frequencies. Table 1 illustrates the 3-tuples representation for the sequences of Fig. 1, where each column corresponds to an unique tuple comprising 3 symbols and the digit in each cell indicates the number of the tuple appearing in the sequence. Based on the table, distance between sequences can be easily computed using Euclidean distance [9], Mahalanobis distance [15], etc. The common clustering methods can also be easily applied to categorical sequences, such as the hierarchical clustering algorithms [9, 10] aimed at organizing sequences into a tree of clusters and the partitioning methods including the well-known K -means and its numerous variants [1, 11, 12]. Since the aim is to generate flat clusters in this paper, we will focus on the latter, that is, grouping sequences according to the occurrences of the tuples given the number of clusters K .

As discussed previously, the number of tuples (i.e., the data dimensionality) would be huge in practice, when the n -tuples representation is employed with the length fixed at a large n . For example, the number of symbols composing a speech sequence typically reaches 20 (see Sect. 4.1); therefore, there is a set of 20^n

Table 1. 3-tuples representation for the sequences shown in Fig. 1.

	AAB	ABA	ABB	ACA	ACB	BAA	BAB	BAC	BBA	CAC	CBA
s_1	1	1	1	1	2	1	0	2	1	1	1
s_2	1	1	1	0	2	1	1	1	1	0	1

possible tuples. To cluster such high-dimensional data, one has to resort to the unsupervised feature-selection techniques, implemented by the filter methods or the built-in methods [1, 12]. Subspace clustering, aimed at grouping data objects into clusters projected in some subspaces, is one of the popular methods using the built-in mechanism for feature selection. Examples include PROCLUS and its variants [16], the entropy-weighting algorithm EWKM [11], etc. The goal of a filter method is to choose an appropriate subset of the original features in the preprocessing step before clustering, where some heuristic criteria are defined to evaluate the significance of features [17]. Due to the huge number of admissible subsets which is exponential in the data dimensionality, usually, both methods choose the subset based on the assumption that the attributes are independent of each other. In the conditional probability model (CPD) for sequences [10], for instance, only the frequent subsequences (corresponding to the tuples) are chosen, given a threshold defining the minimal frequency of the resulting tuples.

In the variable-length representation for the tuples proposed in this paper, however, we focus on the identification of possibly redundant tuples. With the redundant tuples removed, the correlations between features (tuples) are thus reduced. Our efficient method for producing the new representation model is based on the pruning strategy, while surmounting the independent assumption, as described in the next section.

3 Sequences Clustering with Variable-Length Tuples

In this section, we propose a variable-length tuples representation for categorical sequences, followed by a new K -means-type algorithm for clustering the sequences. We begin by introducing the notation used throughout the paper.

3.1 Basic Notation

In what follows, the sequence set is denoted by $S = \{s_i | i = 1, 2, \dots, N\}$ from which K ($1 < K < N$) clusters are searched for. Here, s_i stands for the i th sequence and N the number of sequences to be clustered. Let s be a categorical sequence of length L , where each of the L symbols is one of the categories $\forall x \in X$, with X being the set of symbols and $|X|$ the number of symbols. Moreover, the K clusters are denoted by $c_1, \dots, c_k, \dots, c_K$, each consisting of a disjoint subset of S ; therefore, $S = \cup_{k=1}^K c_k$. The set of K clusters is denoted by $C = \{c_k | k = 1, 2, \dots, K\}$.

A n -length subsequence, also called n -tuples, of sequence $s \in S$ is a segment of n consecutive symbols in s . Note that the length n is also referred to as *memory length* in the case of Markov chain model for sequences [10]. Letting t be a n -length tuple and $\#_S(t)$ the number of t appearing in S , we denote the set of n -tuples by $T = \{t | \#_S(t) > 0\}$; in other words, each of the tuples should appear in at least one of the sequences in S . Based on the definitions, the cardinality of T , i.e., $|T|$, is precisely the dimensionality of the data using the tuples-based representation model. We denote $D = |T|$ and t_d the d th n -tuples of S , where $d = 1, 2, \dots, D$.

Each n -length tuple with $n > 1$ can be viewed as a combination of the preceding $(n - 1)$ -length subsequence and the ending symbol. According to this view, the tuple t can be rewritten as $t = \delta x$, where δ denotes the preceding subsequence of the ending symbol x . We denote the conditional probability of x given its preceding subsequence by $p(\mathcal{X} = x | \mathcal{Y} = \delta)$, where \mathcal{X} and \mathcal{Y} are the random variables associated with the symbols and the preceding subsequences. To simplify the representation, we will use $p(x|\delta)$ to denote $p(\mathcal{X} = x | \mathcal{Y} = \delta)$ in the following pages.

3.2 Variable-Length Tuples Representation

Given a large memory-length n , for example, $n = 10$, the resulting n -tuples representation for sequences is generally in high dimensionality (recall that $D \approx |X|^n$). In this subsection, we aim at reducing the dimensionality by removing the redundant tuples from T . For the purpose, we first organize the n -tuples into a suffix tree, where each path (from the root to one of the leaf nodes of the tree) corresponds to a tuple of length n . Here, the root of the tree is a virtual symbol indicating the beginning of each tuple; thus, the height of the tree is $n + 1$, and the number of children for each node except the leaves is at most $|X|$. Figure 2(b) shows a subtree created for the 3-tuples of the sequences s_1, s_2 in Fig. 1; the tuples used have the same preceding symbol “A”, as Fig. 2(a) shows.

During the creation of the tree, each node (except the root) is attached a value indicating its conditional probability with regard to its preceding

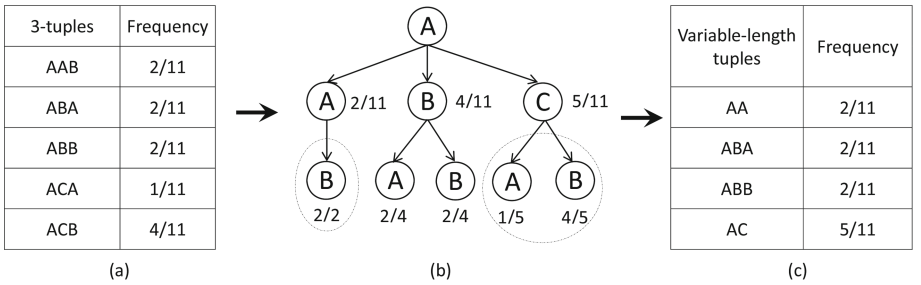


Fig. 2. Illustration of the pruning method for generating the variable-length tuples (preceded “A”) from the sequences shown in Fig. 1

subsequence. The conditional probability is estimated by the frequency estimator. For example, according to Fig. 2(a), the number of the subsequence “AC” appearing in the sequence set $\{s_1, s_2\}$ is $1 + 4 = 5$ and the total number of tuples preceded “A” is 11; then, we estimate the conditional probability by $p(C|A) = 5/11$, as the value attached to the node labeled “C” shows in Fig. 2(b). Likewise, the conditional probabilities for the tuples “ACA” and “ACB” can be estimated, i.e., $p(A|AC) = 1/5$ and $p(B|AC) = 4/5$, respectively.

The variable-length tuples representation for sequences can be derived based on the n -tuples tree. This is achieved by pruning the tree using a post-pruning method similar to that applied to decision-tree induction [1]. As Figs. 2(b) and (c) show, with some leaves removed, the tree changes to accommodate both 2-tuples and 3-tuples. Since we aim at eliminating the redundant tuples, a criterion should be defined to measure the redundancy of a subtree, and subsequently to determine whether the subtree need to prune or not. Taking for example the tree shown in Fig. 2(b) again, obviously, the leaf labeled “B” for the 3-length tuple “AAB” can be deleted, because there is no information loss if it is replaced with the shorter tuple “AA”. However, it is not the case for “AB”: lengthening “AB” to “ABA” and “ABB” is able to obtain considerable information gain. This observation suggests an entropy-based judgement for the redundancy evaluation. Formally, we first compute the entropy for the tuples having the same preceding subsequence δ by

$$H(\delta) = - \sum_{x \in X} p(x|\delta) \times \log_2 p(x|\delta) \quad (1)$$

with

$$p(x|\delta) = \frac{\#_S(\delta x)}{\#_S(\delta)}.$$

Then, the redundant tuples are identified based on the following Definition 1.

Definition 1 (Redundant tuples). The tuples δx for $\forall x \in X$ are redundant if $H(\delta) < \tau$, where $\tau \geq 0$ is a threshold defining the minimal information gain.

Given the n -tuples tree, the pruning process begins by examining the leaf nodes according to Definition 1; here, the symbols corresponding to the leaf and its siblings are considered as $\forall x \in X$ in the senses of Eq. (1). Once they are identified being redundant, the leaves are deleted and their parent node changes to the new leaf. The new leaves are then re-scanned to search for the redundant subsequences of shorter length. With such an iterative pruning method, the 3-tuples for the sequences in Fig. 1 can be reduced into a set of variable-length tuples as shown in Table 2, by setting $\tau = 1$.

Based on the D resulting tuples t_1, t_2, \dots, t_D , we represent each sequence with a D -dimensional vector according to the following Definition 2. Note that each vector \mathbf{V}_s for the sequence s is normalized such that $\|\mathbf{V}_s\| = 1$, where $\|\cdot\|$ denotes the Euclidean norm of a vector. By the normalization, the frequencies are smoothed to counteract the effect induced by the varying lengths of the sequences as well as the tuples.

Table 2. Variable-length tuples representation for the sequences shown in Fig. 1.

	AA	ABA	ABB	AC	BAA	BAB	BAC	BB	CA	CB
s_1	0.224	0.224	0.224	0.670	0.224	0.000	0.447	0.224	0.224	0.224
s_2	0.289	0.289	0.289	0.577	0.289	0.289	0.289	0.289	0.000	0.289

Definition 2 (Variable-length tuples representation). The variable-length tuples representation for each sequence $s \in S$ is the vector \mathbf{V}_s , given by

$$\mathbf{V}_s = \langle f_s(t_1), f_s(t_2), \dots, f_s(t_d), \dots, f_s(t_D) \rangle \quad (2)$$

where $f_s(t_d) = \#_s(t_d) \times (\sum_{d'=1}^D [\#_s(t_{d'})]^2)^{-\frac{1}{2}}$ with $\#_s(t_d)$ being the number of t_d appearing in s .

Now, the only pending factor of the representation model is the setting for τ . Intuitively, the desired value of τ connects to both the number of symbols composing the sequences ($|X|$) and the number of clusters K . In particular, τ should be enlarged with a large $|X|$ and a small K ; thus, an obvious setting for τ could be

$$\tau = \max\{\log_2 \frac{|X|}{K}, 0\}. \quad (3)$$

According to Eq. (3), $\tau > 0$ when $|X| > K$, which is often the case in practice. In the case where $|X| \leq K$, $\tau = 0$ which means that it is not necessary to prune the tuples tree. In this case, the variable-length representation degenerates to the traditional n -tuples representation.

3.3 Clustering Algorithm

In this subsection, a partitioning algorithm is presented for clustering categorical sequences based on the variable-length tuples representation discussed in the previous subsection. The algorithm named *KM-NVLT* (for *K*-Means with Normalized Variable-Length Tuples), as outlined in Algorithm 1, starts clustering from transforming the sequences in S into vectors using the new representation (steps (1)~(3)). Then, the algorithm groups the sequences in an iterative process like the *K*-means clustering (Step (5)).

The aim of Step (4) in *KM-NVLT* is to build a robust condition for the coming iterative process, by choosing a set of well-scattered sequences as the initial cluster centers. The first two centers I_1 and I_2 are chosen according to the following rule:

$$(I_1, I_2) = \operatorname{argmax}_{(s, s') \in S \times S} \|\mathbf{V}_s - \mathbf{V}_{s'}\|^2. \quad (4)$$

Then, the remaining $K - 2$ centers are selected based on the maximum-minimum principle [9], i.e.,

$$I_{k+1} = \operatorname{argmax}_{s \in S \setminus \{I_i | i=1, \dots, k\}} \min_{i=1, \dots, k} \|\mathbf{M}_s - \mathbf{M}_{I_i}\|^2 \quad (5)$$

where $k \in [2, K - 1]$.

Input: the sequence set S , the number of clusters K and the memory-length n
Output: the set of resulting clusters $C = \{c_k | k = 1, 2, \dots, K\}$
begin
 (1) Generate n -tuples from the sequences in S , and create the n -tuples tree using the method described in Sect. 3.2;
 (2) Determine τ according to Eq. (3) and prune the n -tuples tree based on Definition 1;
 (3) Collect the variable-length tuples from the pruned tree, and represent each sequence $s \in S$ by the vector \mathbf{V}_s according to Eq. (2) and Definition 2;
 (4) Choose K vectors for the initial cluster centers using Eqs. (4) and (5);
 (5) **repeat**
 (5.1) Generate c_1, c_2, \dots, c_K by assigning each sequence $s \in S$ to its closest cluster center, in terms of the Euclidean distance between \mathbf{V}_s and each cluster center;
 (5.2) Recompute the center for each cluster c_k by averaging the vectors belonging to c_k , where $k = 1, 2, \dots, K$.
 until C is not changed.;
end

Algorithm 1. Outline of the *KM-NVLT* algorithm.

The time complexity for generating the variable-length tuples is $O(n \times |X| \times \mathcal{L})$, where \mathcal{L} is the total lengths of the sequences in S . The time complexities of the K -means-type clustering and the centers selection method are $O(KND)$ and $O(N^2D)$, respectively. Generally, $|X| > K$ and $n \times \mathcal{L} > ND$; thus, for *KM-NVLT*, the time complexity can be finally given as $O(\max\{n \times |X| \times \mathcal{L}, N^2D\})$.

4 Experimental Evaluation

In this section, we evaluate the performance of *KM-NVLT* on real-world categorical sequences, and also experimentally compare the variable-length tuples representation with a few other methods.

4.1 Sequence Sets and Experimental Setup

Six sequence sets for speech recognition are used. We obtained the sequence sets from [18], namely *locmelovoy*, *locmrlovoy*, *locmslovoy*, *locfjlavoy*, *locflavoy* and *locffpevoy*, abbreviated to S1 ~ S6, respectively. Each sequence in the sets is generated from the pronunciation of one of the five French vowels (“a”, “e”, “i”, “o” and “u”) by binning the sound wave. So, the true number of clusters for the sequence sets are known, i.e., $K = 5$. Details of the data sets are summarized in Table 3. The average length of sequences in these sets varies from 560 to approximately 1900, in order to evaluate the capability of different representation models and clustering methods.

We used the K -means algorithm on the common n -tuples representation as the baseline in the experiments. The algorithm is abbreviated KM-NT (for K -Means with Normalized Tuples), which was implemented as a special case of our

Table 3. Details of the real-world sequence sets

Dataset	#clusters(K)	#symbols($ X $)	#sequences(N)	Length(L)	Average length
S1	5	20	50	[203, 1035]	560
S2	5	18	50	[226, 1253]	737
S3	5	20	50	[506, 1564]	924
S4	5	19	50	[405, 1986]	1088
S5	5	19	50	[581, 2382]	1570
S6	5	18	50	[701, 3753]	1899

KM-NVLT by fixing τ at 0. Based on the normalized n -tuples representation, we also chose EWKM [11], a feature-weighting-based subspace clustering algorithm, as the competing method. Its weighting parameter γ was set to the author-recommended value 0.5.

Frequency-based feature selection method was also used to provide a reference point for comparison. For the purpose, the frequent n -tuples were collected to create a reduced tuples set for each sequence set. Since it is difficult to determine the threshold for the frequencies, we selected those tuples whose frequency is larger than 1 for the resulting representation. The vectors were finally normalized using the same method to that of Definition 2. We will denote the K -means algorithm applied to such a representation by KM-NFT (for K -Means with Normalized Frequent Tuples).

Bisection K -means [9] was also chosen for comparison. For this algorithm, we created a TF-IDF representation model for each sequence set, where IDF is the abbreviation for the *inverse document frequency* popularly used in the text mining community [7]. After assigning the n -tuples with the IDF weights, the vectors were normalized. The bisection K -means with the TF-IDF representation will be denoted as BKM-NIDF. The initial cluster centers for all the competing algorithms were selected using the same approach as that of *KM-NVLT* (see Step (4) of Algorithm 1).

4.2 Experimental Results

The performance of the algorithms is evaluated in terms of *clustering accuracy* (CA), which is computed as $CA = \frac{1}{N} \sum_{k=1}^K a_k$, where a_k is the number of sequences in the majority group corresponding to c_k . Clearly, this measure requires that the ground truth of the datasets be known, which is the case in our experiments. Table 4 shows the clustering accuracy obtained by the five algorithms on the six sequence sets, with the memory-length n set to 10 (the performance with regard to different n will be examined in Fig. 3). The best results are marked in bold typeface.

From Table 4, we can see that our *KM-NVLT* is able to achieve high-quality overall results, outperforming the competing algorithms on all the six sequence sets. In fact, only KM-NFT obtains comparable results on S1, S3 and S6, whereas

Table 4. Comparisons of clustering accuracy (CA) obtained by different algorithms ($n = 10$)

Dataset	<i>KM-NVLT</i>	KM-NFT	EWKM-NT	BKM-NIDF	KM-NT
S1	1.000	0.980	0.620	0.720	0.620
S2	0.980	0.600	0.600	0.700	0.600
S3	1.000	1.000	0.660	0.640	0.660
S4	0.860	0.640	0.680	0.720	0.680
S5	1.000	0.780	0.620	0.680	0.620
S6	0.980	0.900	0.680	0.660	0.700

EWKM-NT and KM-NT perform poorly (note that both are based on the n -tuples representation). We also observe that the clustering accuracies of EWKM-NT and KM-NT are close, indicating that the built-in feature-selection scheme used in the soft subspace clustering methods fails in distinguishing between significant and non-significant tuples. BKM-NIDF, which makes use of the TF-IDF representation, yields higher accuracy than EWKM-NT on most of the data sets. This indicates that the IDF weighting method is able to identify significant tuples to some extent.

Comparisons of the clustering accuracy with varying memory-length of tuples are given in Fig. 3. It can be seen from the figures that our *KM-NVLT* achieves robust performance accompanied by high clustering accuracy, along with the increment of the memory-length n . Except the cases of KM-NFT on S1 and S3, the competing algorithms yield instable results that are sensitive to the setting of n . The good performance of *KM-NVLT* owes to the use of the pruning method in producing the variable-length tuples with redundant tuples deleted, which, in effect, improve the performance of the remaining features (tuples). Another gain of the pruning method is to reduce the number of features for the clustering algorithms, as Fig. 4 shows.

Figure 4 illustrates the number of resulting tuples in the traditional n -tuples representation (used by EWKM-NT and KM-NT), the frequent n -tuples representation used by KM-NFT and our variable-length representation. One can see that the number of tuples are significantly reduced by using our pruning method. The number can also be reduced using the frequency-based selection method; however, the results are clearly dependent on the user-defined threshold, which is difficult to estimate. The figures also show that, when the memory-length n goes from 6 to 14, the numbers of resulting variable-length tuples remain approximately unchanged on the six sequence sets. This result suggests that the optimal length of tuples for the speech sequences is about 6. As the memory length substantially connects to the order of Markov chain model [10], our pruning method might be helpful in estimating the order for such models.

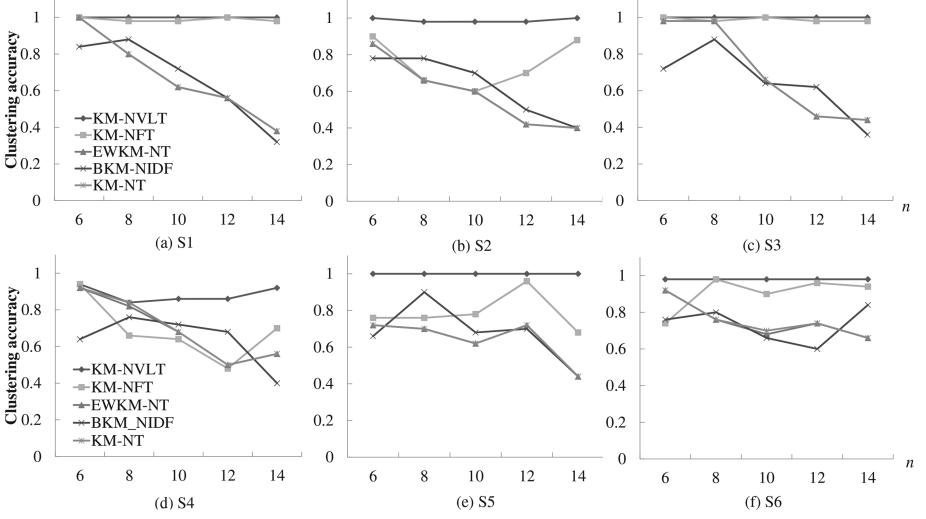


Fig. 3. Clustering accuracy of the algorithms with various memory-lengths of tuples.

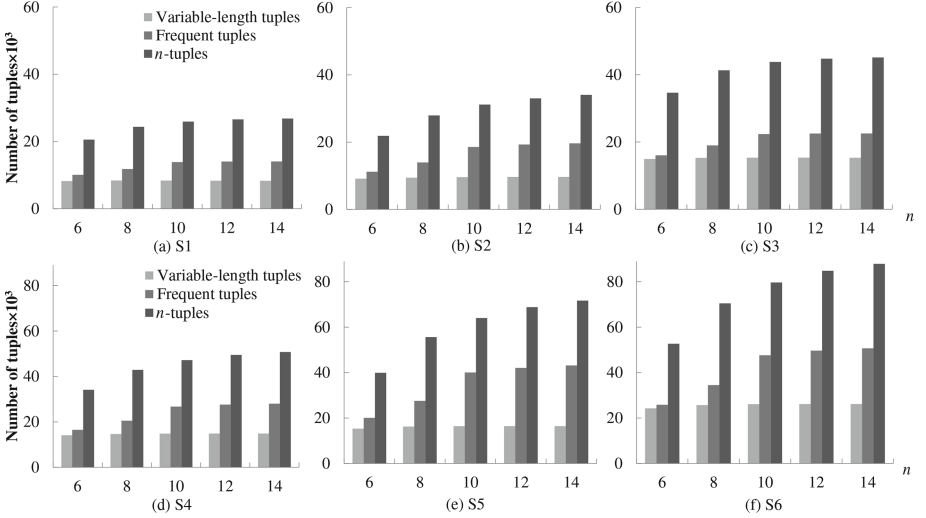


Fig. 4. Comparisons of the number of tuples using different representation models.

5 Conclusion and Perspectives

In this paper, we proposed a variable-length tuples representation for categorical sequence clustering, unlike the existing methods, which are generally based on the fixed-length tuples (n -tuples) representation. We proposed to organize the original n -tuples into a tree, in order to derive a pruning method to obtain the variable-length tuples from the pruned tree. We defined an entropy-based

measure to evaluate the redundance of tuples and to provide the basis for the removal of redundant tuples. Using the resulting variable-length tuples, we also proposed a *K*-means-type algorithm, call *KM-NVLT*, for categorical sequences clustering. The experiments were conducted on six speech sequence sets, the results show the effectiveness of the new representation for sequences and the new algorithm for clustering.

There are many directions that are clearly of interest for future exploration. One avenue of further study is to test *KM-NVLT* on more extensive sequence sets, and to compare with other mainstream methods. Another efforts will be directed towards extending the method to variable-order Markov chain model for model-based categorical sequence clustering.

Acknowledgments. This work was supported by the National Natural Science Foundation of China under Grant No. 61175123, and partially supported by the Natural Science Foundation of Fujian Province of China under Grant No. 2015J01238.

References

1. Aggarwal, C.C.: Data Mining: The Textbook. Springer, New York (2015)
2. Xu, R., Wunsch, D.C.: Survey of clustering algorithms. *IEEE Trans. Neural Netw.* **16**, 645–678 (2005)
3. Yang, J., Wang, W.: CLUSEQ: Efficient and effective sequence clustering. In: Proceedings of IEEE ICDE, pp. 101–112 (2003)
4. Dong, G., Pei, J.: Classification, clustering, features and distances of sequence data. *Seq. Data Min.* **33**, 47–65 (2007)
5. Kelil, A., Wang, S.: SCS: a new similarity measure for categorical sequences. In: Proceedings of IEEE ICDM, pp. 343–352 (2008)
6. Vingá, S., Almeida, J.: Alignment-free sequence comparison: a review. *Bioinformatics* **19**, 513–523 (2003)
7. Leopold, E., Kindermann, J.: Text categorization with support vector machines: how to represent texts in input space? *Mach. Learn.* **46**, 423–444 (2002)
8. Kondrak, G.: *N*-Gram similarity and distance. In: Consens, M., Navarro, G. (eds.) SPIRE 2005. LNCS, vol. 3772, pp. 115–126. Springer, Heidelberg (2005). doi:[10.1007/11575832_13](https://doi.org/10.1007/11575832_13)
9. Wei, D., Jiang, Q., Wei, Y., Wang, S.: A novel hierarchical clustering algorithm for gene sequences. *BMC Bioinform.* **13**, 174 (2012)
10. Xiong, T., Wang, S., Jiang, Q., Huang, J.Z.: A novel variable-order Markov model for clustering categorical sequences. *IEEE Trans. Knowl. Data Eng.* **26**, 2339–2353 (2014)
11. Jing, L., Ng, M.K., Huang, J.Z.: An entropy weighting k-means algorithm for subspace clustering of high-dimensinoal sparse data. *IEEE Trans. Knowl. Data Eng.* **19**, 1–16 (2007)
12. Chen, L., Jiang, Q., Wang, S.: Model-based method for projective clustering. *IEEE Trans. Knowl. Data Eng.* **24**, 1291–1305 (2012)
13. Herranz, J., Nin, J.: Solé M.: optimal symbol alignment distance: a new distance for sequences of symbols. *IEEE Trans. Knowl. Data Eng.* **23**, 1541–1554 (2011)
14. Chen, L.: EM-type method for measuring graph dissimilarity. *Int. J. Mach. Learn. Cybern.* **5**, 625–633 (2014)

15. Wu, T.J., Burke, J.P., Davison, D.B.: A measure of DNA sequence dissimilarity based on Mahalanobis distance between frequencies of words. *Biometrics*. **53**, 1431–1439 (1997)
16. Wu, T., Fan, Y., Hong, Z., Chen, L.: Subspace clustering on mobile data for discovering circle of friends. In: Zhang, S., Wirsing, M., Zhang, Z. (eds.) KSEM 2015. LNCS (LNAI), vol. 9403, pp. 703–711. Springer, Heidelberg (2015). doi:[10.1007/978-3-319-25159-2_64](https://doi.org/10.1007/978-3-319-25159-2_64)
17. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**, 1226–1238 (2005)
18. Loisel, S., Rouat, J., Pressnitzer, D., Thorpe, S.: Exploration of rank order coding with spiking neural networks for speech recognition. *Proc. IEEE IJCNN* **4**, 2076–2080 (2005)

Knowledge Science, Engineering and Management
9th International Conference, KSEM 2016, Passau,
Germany, October 5-7, 2016, Proceedings
Lehner, F.; Fteimi, N. (Eds.)
2016, XIX, 642 p. 216 illus., Softcover
ISBN: 978-3-319-47649-0