An Occurrence Based Approach to Mine Emerging Sequences

Kang Deng and Osmar R. Zaïane

Department of Computing Science, University of Alberta
Edmonton, Alberta, T6G 2E8
{kdeng2,zaiane}@ualberta.ca

Abstract. An important purpose of sequence analysis is to find the distinguishing characteristics of sequence classes. Emerging Sequences (ESs), subsequences that are frequent in sequences of one group and less frequent in the sequences of another, can contrast sequences of different classes and thus facilitating sequence classification. Different approaches have been developed to extract ESs, in which various mining criterions are applied. In our work we compare Emerging Sequences fulfilling different constraints. By measuring ESs with their occurrences, introducing gap constraint and keeping the uniqueness of items, our ESs demonstrate desirable discriminative power. Evaluating against two mining algorithms based on support and no gap constraint subsequences, the experiments on two types of datasets show that the ESs fulfilling our selection criterions achieve a satisfactory classification accuracy: an average F-measure of 93.2% is attained when the experiments are performed on 11 datasets.

Keywords: Emerging Sequences, Classification, Occurrence Count.

1 Introduction

Sequence comparison is an significant Data Mining Task [4], where the distinguishing subsequences play an important role in the contrast. Given two sequence groups, Emerging Sequences (ESs) is defined as subsequences that are frequent in sequences of one group and less frequent in the sequences of another, and thus distinguishing or contrasting sequences of different classes [3]. With the discriminative power of emerging sequences, prediction models trained by using ESs perform well and achieve satisfactory classification accuracies on labeling sequence instances.

Different approaches have been developed to extract ESs, in which various mining criterions are applied. For instance, in bioinformatics, researchers align genome by using substrings [7], in which, items have to appear immediately next to each other in the original sequence. However, Lo et al. [8] removed the restriction that related events must occur close together in a sequence, i.e. the distance between two events could be arbitrarily large. Furthermore, most sequential mining algorithms regard support of features (the number of transactions that contain the feature) as the selection standard, while the total occurrence of the feature is also crucial.

In our research, we compare Emerging Sequences fulfilling different constraints and try to find the important factors for ESs. Besides the frequency distinction, we discover that the following criterions are also significant in the sequence classification:

– Occurrences of subsequences are more informative than supports.
– Items are unique (i.e. not repeated) in a given subsequence.
– Any two adjacent items in the subsequence should be close in the original sequence.

To mine subsequences fulfilling the above conditions, we provide an algorithm as well as a pruning strategy based on a previous work [6]. After ES candidates are extracted, we perform feature selection by F-ratio [10], by which the distinguishing subsequences are selected to represent the original sequence groups. To evaluate the discriminative power of emerging sequences, a SVM classifier [2] is trained by using ESs to classify sequence instances. In this learning framework, higher prediction accuracy indicates better emerging sequences.

For comparison, we perform controlled experiments on two recent and well-known sequence mining algorithms. One algorithm, ConsGapMiner [6] can control the gap between related events, while another algorithm [8] removes the gap constraint and extracts iterative patterns. We choose two types of datasets, one is the UNIX user command sequences [1], the other is a software behavior history [9].

The experiments demonstrate the effectiveness of our emerging sequences: the classifier based on ESs outperforms the other two baseline algorithms. The prediction accuracy measured by the average F-measure is 93.2% when the experiments are performed on 11 datasets. On the datasets CVS-Omission and MySQL [9], our model perfectly labels the sequences with a prediction accuracy of 100%.

In the next section, we introduce some terminology. In Section 3, we describe the sequence mining algorithm and the feature selection strategy. We present the prediction performance of our proposed approach in Section 4. Finally, Section 5 presents our conclusions.

2 Preliminaries

Let $I = \{i_1, i_2, \ldots, i_k\}$ be a set of all items, or the alphabet, a sequence is an ordered list of items from I. Given a sequence $S = \langle s_1, s_2, \ldots, s_m \rangle$ and a sequence $S' = \langle s'_1, s'_2, \ldots, s'_n \rangle$, we say that S' is a subsequence of S or S contains S', denoted as $S' \sqsubseteq S$, if there exist integers $1 \leq j_1 < j_2 < \ldots < j_n \leq m$ such that $s'_1 = s_{j_1}$, $s'_2 = s_{j_2}$, …, $s'_n = s_{j_n}$.

Definition 1 (Subsequence Occurrence). *Given a sequence $S = \langle s_1, \ldots, s_n \rangle$ and a subsequence $S' = \langle s'_1, s'_2, \ldots, s'_m \rangle$ of S, an occurrence of S' is a sequence of indices $\{i_1, i_2, \ldots, i_m\}$, whose items represent the positions of elements in S.*

For instance, if sequence $S = \langle B, C, B, C, A, C \rangle$, and its subsequence $S' = \langle B, C \rangle$. There are 5 occurrences of S' in S: $\{1, 2\}$, $\{1, 4\}$, $\{1, 6\}$, $\{3, 4\}$ and $\{3, 6\}$.

Definition 2 (Gap Constraint). *The gap constraint is specified by a positive integer g. In a subsequence occurrence $o_s = \{i_1, i_2, i_3, \ldots, i_m\}$, the difference of any two adjacent indices is $i_{k+1} - i_k$. If $i_{k+1} - i_k \leq g + 1$, we say the occurrence o_s fulfills the g-gap constraint.*

For example, if $g = 1$, the occurrences of S' $\{1, 2\}$ and $\{3, 4\}$ fulfill the 1-gap constraint (also 0-gap) but $\{1, 4\}$, $\{1, 6\}$ and $\{3, 6\}$ do not.

Definition 3 (Support and Occurrence Count). *Given a sequence dataset $\mathcal{D}_c$, where c is a class label, $\mathcal{D}_c$ consists of a set of sequences. The support of a subsequence α is the number of sequences in $\mathcal{D}_c$ that contain α, while the occurrence count is the number of non-overlapping occurrences of α in $\mathcal{D}_c$.*

For example, in Table 1, if the gap constraint is 1, the support of the sequence $\alpha = \langle a, b \rangle$ in $\mathcal{D}_{pos}$ is 3, meaning all sequences contain α while fulfilling 1-gap constraint. The occurrence count of α is 4, because α appears twice in Sequence 1. One thing we need to notice is that the total occurrences of α fulfilling 1-gap constraint in Sequence 1 is 5. However, some of them are overlapped, so the non-overlapping count is 2.

In this paper, related support and count, denoted as $support(\alpha, \mathcal{D}_c)$ and $count(\alpha, \mathcal{D}_c)$ respectively, are used to measure the frequency of subsequences. As for the example above, $support(\alpha, \mathcal{D}_c) = \frac{3}{3}$ and $count(\alpha, \mathcal{D}_c) = \frac{4}{3}$.

Table 1. A sequence dataset example

sequence ID	sequences	labels
1	*aabbcab*	*pos*
2	*cadb*	*pos*
3	*bcab*	*pos*
4	*acabd*	*neg*
5	*bda*	*neg*

The notion of Emerging Sequences (ESs) was introduced by Zaïane et al. [11], here we generalize this notion and define:

Definition 4 (Emerging Sequences). *Given two contrasting sequence classes, Emerging Sequences (ESs) are subsequences that are frequent in sequences of one group and less frequent in the sequences of another, and thus distinguishing or contrasting sequences of different classes.*

3 Sequence Mining and Feature Selection

To distinguish one group of sequence data from another, representative subsequences must be extracted. In this section, we explain how we first extract the ES candidates; then implement a dynamic feature selection to mine the most discriminative subsequences.

3.1 Mining Criterion

To mine the representative subsequences, one fundamental question is: "what kind of sequences should we choose?" An essential selection criterion is that features should be discriminative. Let $\mathcal{D}_{pos}$ and $\mathcal{D}_{neg}$ be two classes of sequences; the occurrence counts of a ES candidate α in both classes, denoted as $count(\alpha, \mathcal{D}_{pos})$ and $count(\alpha, \mathcal{D}_{neg})$, need to meet the following conditions:

$$count(\alpha, \mathcal{D}_{pos}) > \theta \tag{1}$$

$$count(\alpha, \mathcal{D}_{neg}) \le \theta \tag{2}$$

where θ is the minimum count threshold.

Instead of supports, we use the occurrence counts of subsequences to measure their discriminative power, because repetitive features within a sequence is important. For instance, a UNIX user may repeatedly type the same command pattern within one session.

Another mining principle we apply is that items are unique in one sequence pattern. Since the multiple occurrences of patterns in each original sequence are counted, it is not necessary to consider subsequences with repetitive items.

The last standard is that items have to appear closely with each other in the original sequence, as items far apart are less relevant in the decision making. An example is the relationships between words in a long sentence. A verb probably serves as the predicate of a subject if they are close to each other. Therefore, gap constraints need to be considered in subsequence mining.

3.2 ES Candidates Extraction

To control the gap constraint when mining emerging sequences, our mining model is based on a previous work, ConsGapMiner [6]. In their approach, however, they choose support as the selection criterion, and items are not unique in the sequence pattern.

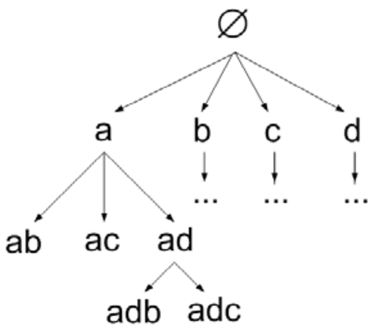

Fig. 1. Candidate Generation Tree

We enumerate ES candidates by a Depth First Search. Given an ES candidate, an item from the vocabulary is appended at the ending of the current subsequence, so a new candidate is generated. Figure 1 shows the candidate generation tree. Given a vocabulary $I = \{a, b, c, d\}$, the root of the tree is an empty set. For a subsequence $\langle a \rangle$, if it fulfills the discriminative conditions 1, 2 and the gap constraint, three new candidates are generated by appending b, c and d respectively. Sequence pattern $\langle a, a \rangle$ is not taken into consideration, because items ought to be unique in our emerging sequences. This pruning strategy can greatly reduce the searching space and improve the scalability of the tree generation algorithm.

3.3 Support Calculation

Given an ES candidate, the next problem is to validate if the candidate fulfills the discriminative conditions and the gap constraint. In [6], a bitset operation-based algorithm is proposed because the bit is the basic operation unit in computers.

A bitset is a sequence of bits which each takes the value 0 or 1, indicating the subsequence occurrences in the original sequence. The bitset of the length-1 subsequence is easy to generate, i.e. 1 indicates the appearance of the item, while other digits are 0s. Given the sequence $\langle aabbcab \rangle$, the bitset of the subsequence $\langle a \rangle$ is simply 1100010. And the occurrence count of a length-1 subsequence is also straightforward: it is the number of 1s in the bitset and the gap constraint is irrelevant. Therefore, the occurrence count of $\langle a \rangle$ is 3.

For a subsequence whose length is larger than 1, the calculation of its occurrence count is more complicated, because the gap constraint is taken into account. It has three steps:

1. Perform right shift operation on the bitset of its parent to generate the mask bitset.
2. Attain the subsequence bitset by **AND** operation.
3. Calculate the occurrence count based on the bitset.

The first step is to generate the mask bitset by the parent of the target subsequence. From the last example, given a subsequence $\alpha = \langle a, b \rangle$, the bitset of its parent $\alpha_p = \langle a \rangle$ is known. To calculate the mask bitset when the gap constraint is g, we right shift the bitset of α_p for $g + 1$ times, then perform **OR** operation on the results. As the bitset of α_p is 1100010, the process is as follows:

$$\begin{array}{r}
1100010 >> 0110001 \\
0110001 >> 0011000 \\
\hline
\textbf{OR} \quad >> 0111001
\end{array}$$

So the mask bitset is 0111001.

Based on the mask bitset and the bitset of the last item of α, the bitset of α is generated by **AND**ing them. Taking the last example, the mask bitset is 0111001 and the bitset of $\langle b \rangle$ is 0011001, by **AND**ing them:

$$\begin{array}{r} 0111001 \\ 0011001 \\ \hline \textbf{AND}\ 0011001 \end{array}$$

the bitset of $\alpha = \langle a, b \rangle$ is 0011001.

Finally, it is the calculation of the occurrence count. Since there is more than one item in the subsequence, we cannot simply count the number of 1s in the bitset. Given a bitset and a gap constraint g, for each 1 in the bitset, the following g digit(s) must be set to 0. Then the occurrence count is the number of 1s in the bitset. For the last example, the bitset 0011001 is converted to 0010001, so the occurrence count of the pattern $\alpha = \langle a, b \rangle$ in the sequence $\langle aabbcab \rangle$ is 2.

3.4 Feature Selection

By combining the candidate generation tree and the bitset operation, numerous ES candidates are extracted. In this subsection, we refine the result and select the most discriminative subsequences as ESs.

Given several sequence groups $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_m\}$ and a set of subsequences $\{s_1, s_2, \ldots, s_k\}$, the objective is to find the most discriminative subsequences. For an ideal emerging sequence, its occurrence counts in several groups should differ greatly, i.e. the variance between groups should be much larger than that within each group. To solve this classic analysis of variance (ANOVA) problem, we apply F-ratio to measure the discriminative power:

$$F - ratio = \frac{MS_{between}}{MS_{within}} \tag{3}$$

where $MS_{between}$ is the mean square (variance estimate) explained by the different groups, and MS_{within} is mean square (variance estimate) that is due to chance (unexplained). Given the number of groups m and the total number of sequences N, $MS_{between}$ and MS_{within} are defined as:

$$MS_{between} = \frac{\sum_i n_i(\overline{c_i} - \overline{c})^2}{m-1} \tag{4}$$

$$MS_{within} = \frac{\sum_{ij}(c_{ij} - \overline{c_i})^2}{N-m} \tag{5}$$

where n_i is the number of sequences in group i, c_{ij} is the occurrence count in the jth sequence of the ith group, $\overline{c_i}$ is the mean of the occurrence counts in Group i, and $\overline{c}$ is the mean of those for all samples. As m and N are fixed for all subsequences, the F-ratio can be simplified as:

$$F - ratio = \frac{\sum_i n_i(\overline{c_i} - \overline{c})^2}{\sum_{ij}(c_{ij} - \overline{c_i})^2} \tag{6}$$

From Equation 6, we can see that, for an ES candidate, when the variance of the occurrence count between groups is large and that within groups is small, its F-ratio become large. Based on the F-ratio, we rank the ES candidates, and the highly-ranked ones are more discriminative.

To avoid numerous ESs, we then perform a dynamic feature selection strategy [5], i.e. only the top-m subsequences, based on F-ratio, are kept. It guarantees that each sequence can be represented by at least m ESs (the high-ranked ones) and the database does not become too large due to the possible sheer number of candidate subsequences.

4 Experimental Results

In the last section, emerging sequences are selected by our occurrence count based mining framework. To verify the discriminative power of ESs, we then perform the controlled experiments on ESs.

4.1 Evaluation Methodology

To perform the experiments, the sequence datasets are transformed to transactional datasets in order to be in a suitable form for learning algorithms, i.e. each sequence is represented by a set of attribute-value pairs, where the attribute represents an emerging sequence, and the value is its occurrence count in this sequence. Then, a classifier is trained by using the transactional datasets. In this paper, we choose a well-developed classification package LIBSVM [2] as the prediction model. Finally, we perform a 6-folder cross validation on the classification framework. The average prediction accuracy, represented by f-measures, indicates the performance of the selected features.

For comparison, we chose two other recent and well-known mining algorithms to extract different features from the original datasets:

– Minimal Distinguishing Subsequences (MDSs): this kind of sequences are mined by ConSGapMiner [6]. There are two main differences between MDSs and our ESs: 1. they use support as the selection criterion, while occurrence count is applied in our mining model. 2. Items are unique in ESs but can be repetitive in MDSs.
– Iterative Patterns (IPs) [8]: IPs achieve satisfactory performance on classifying software behaviour sequences. As opposed to our ESs, they remove the restriction that related events must occur close together in a sequence, i.e. the distance between two events could be arbitrarily large.

These two kinds of features are selected and used in the validation framework above. By comparing the prediction accuracies of those features, we can verify the effectiveness of our selection criterions:

– Occurrences of subsequences are more informative than supports.
– Items are unique in the subsequences (i.e. not repeated).
– Any two adjacent items in the subsequence should be close in the original sequence.

4.2 UNIX User Command Dataset

The first type of datasets we use is the UNIX user commands dataset from the UCI Machine Learning Repository [1]. It contains 9 sets of sanitized user data drawn from the command histories of 8 UNIX computer users at Purdue University. This dataset only keeps command names, flags, and shell meta characters, while removing filenames, user names, directory structures etc. For each user, we select 100 sequences. In each experiment, two users' commands are chosen, and the F-measures and standard deviations are presented in Table 2.

Table 2. Classification performances on the UNIX dataset

Datasets	Length	Size	MDSs	IPs	ESs
user 0 and 1	28	176	0.770 ± 0.053	0.776 ± 0.065	$\mathbf{0.806} \pm 0.048$
user 1 and 8	33	268	0.938 ± 0.045	$\mathbf{0.959} \pm 0.024$	0.958 ± 0.035
user 2 and 3	39	231	$\mathbf{0.970} \pm 0.018$	0.948 ± 0.078	0.965 ± 0.032
user 3 and 6	36	231	0.918 ± 0.058	0.913 ± 0.044	$\mathbf{0.929} \pm 0.044$
user 5 and 6	49	278	0.865 ± 0.092	0.865 ± 0.060	$\mathbf{0.903} \pm 0.062$
user 5 and 7	44	284	0.908 ± 0.028	0.905 ± 0.031	$\mathbf{0.920} \pm 0.029$

In Table 2, column Length represents the average sequence length of this user pair, while column Size means the vocabulary size. We observe that the prediction accuracy of our ESs-based approach is comparable or better than the other two features. This demonstrates that our selection criterions are effective with various average sequence length and vocabulary size.

4.3 Software Behaviour Dataset

The second type of datasets is the set of software behaviour sequences. Software behavior is the way a program executes. From the start of the program until its termination, the execution events are recorded. A software behaviour, composed by a sequence of normal individual execution events, could be broken down by their interaction in an undesirable order. Therefore, the objective of the analysis is to distinguish deviant software behaviours from regular ones by sequence mining.

Lo et al. [8] focus on this type of data and proposed Iterative Patterns (IPs) mining algorithm, which achieves satisfactory performance and improves the prediction accuracy greatly. So in this subsection, we focus on the comparison between IPs and ESs. The same datasets as in [8] are chosen to perform the experiment. For more information about the dataset, please refer to [8].

Table 3 presents the comparison between the IPs-based and the ESs-based SVM classifiers. Compared with IPs which was designed specifically for software behaviour datasets, our emerging sequences also achieve satisfactory classification accuracy.

Table 3. Classification performances on the software behaviour dataset

Datasets	Length	Size	IPs	ESs
CVS-Mix	9	16	0.935 ± 0.060	$\mathbf{0.945} \pm 0.058$
CVS-Omission	10	16	1 ± 0	1 ± 0
CVS-Ordering	9	16	0.857 ± 0.031	$\mathbf{0.951} \pm 0.032$
MySQL	24	16	1 ± 0	1 ± 0
X11	4	8	$\mathbf{0.979} \pm 0.015$	0.888 ± 0.024

5 Conclusion

In this paper, we focus on Emerging Sequences (ESs), which are frequent in sequences of one group and less frequent in the sequences of another, and thus distinguishing or contrasting sequences of different classes. After comparing ESs of various characteristics, we find that: 1. the occurrence count can measure the discriminative power of ESs more precisely; 2. the uniqueness of items in a subsequence is important; 3. the gap constraint is relevant in the decision making. A mining model is proposed to extract ESs fulfilling our selection criterions.

The experiments demonstrate the effectiveness of our emerging sequences: an average F-measure of 93.2% is achieved when the experiments are performed on 11 datasets. In the datasets CVS-Omission and MySQL [9], our model perfectly labels the sequences with a prediction accuracy of 100%.

However, since our candidate generation algorithm is based on the Depth First Search Tree, the scalability of our current approach is not desirable. As a future work, we are looking into the possibility of a more efficient mining algorithm or pruning strategies, while preserving and potentially improving the prediction accuracy of the classification model.

Acknowledgement

The execution program and datasets for the iterative patterns were provided by the original author, Dr. Lo et al. We would like to acknowledge their help in this regard.

References

1. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
2. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
3. Deng, K., Zaïane, O.R.: Contrasting sequence groups by emerging sequences. Discovery Science, 377–384 (2009)
4. Han, J., Kamber, M.: Data Mining, Concepts and Techniques. Morgan Kaufmann, San Francisco (2001)
5. Jazayeri, S.V., Zaïane, O.R.: Plant protein localization using discriminative and frequent partition-based subsequences. In: ICDM Workshops, pp. 228–237 (2008)

6. Ji, X., Bailey, J., Dong, G.: Mining minimal distinguishing subsequence patterns with gap constraints. Knowl. Inf. Syst. 11(3), 259–286 (2007)
7. Kurtz, S.: Reputer: fast computation of maximal repeats in complete genomes (1995)
8. Lo, D., Cheng, H., Han, J., Khoo, S.-C.: Classification of software behaviors for failure detection: A discriminative pattern mining approach. In: KDD (2009)
9. Lo, D., Cheng, H., Han, J., Khoo, S.-C.: Technical report, School of Information Systems, Singapore Management University (2009), `http://www.mysmu.edu/faculty/davidlo/kdd09.htm`
10. Lomax, R.G., Hahs-Vaughn, D.L., Lomax, R.G.: Statistical Concepts: A Second Course, 3rd edn. Routledge, New York (2007)
11. Zaïane, O.R., Yacef, K., Kay, J.: Finding top-n emerging sequences to contrast sequence sets. Technical Report TR07-03, Department of Computing Science, University of Alberta (February 2007)