Time-Sensitive Feature Mining for Temporal Sequence Classification

Yong Yang, Longbing Cao, and Li Liu

Data Sciences & Knowledge Discovery Lab,
Center for Quantum Computation and Intelligent Systems,
Faculty of Engineering & Information Technology,
University of Technology, Sydney
{yongyang,lbcao,liliu}@it.uts.edu.au

Abstract. Behavior analysis received much attention in recent year, such as customer-relationship management, social security surveillance and e-business. Discovering high impact-driven behavior patterns is important for detecting and preventing their occurrences and reducing resulting risks and losses to our society. In data mining community, researchers pay little attention to time-stamps in temporal behavior sequences (without explicitly considering inherent temporal information) during classification. In this paper, we propose a novel Temporal Feature Extraction Method - TFEM. It extracts sequential pattern features where each transition is annotated with a typical transition time (its duration or interval). Therefore it substantially enriches temporal characteristics derived from temporal sequences, yielding improvements in performances, as demonstrated by a set of experiments performed on synthetic and real-world datasets. In addition, TFEM has the merit of simplicity in implementation and its pattern-based architecture can generate human-readable results and supply clear interpretability to users. Meanwhile, it is adjustable and adaptive to user's different configurations, allowing a tradeoff between classification accuracy and time cost.

1 Introduction

Behavior analysis [1,2] is increasingly regarded as a key component in business problem-solving. Unlike traditional analytical methods, behavior informatics is aimed at discovering high impact events (i.e. those activities associated with or causing a specific impact of interest to the business world) from behavioral data. Discovering high impact-driven behavior patterns is important for detecting and preventing their occurrences and reducing resulting risks and losses to our society, such as earthquake prediction, epidemic outbreak monitoring, market surveillance, fraud detection and national security. In order to identify high impact behavior patterns, the usual transactional data needs to be converted into behavioral data, which is organized to explicitly present properties associated with behavior and its impact on business.

A typical situation of recording behavior is through constructing sequences of behavior, and generating so-called sequential data. Sequential data is widely seen in many applications, including business applications and scientific applications. In general, sequential data only involves the ordering relationship existing in behavior sequences.

B.-T. Zhang and M.A. Orgun (Eds.): PRICAI 2010, LNAI 6230, pp. 315–326, 2010.

Table 1. An example dataset of sequences with timestamps

ID	t_1	t_2	t_3	t_4	t_5	t_6	...	$label$
s_1	a	c	(bd)	c	b	(ac)	...	c_1
s_2	b	a	a	a	a	b	...	c_2
s_3	c	a	a	a	a	(ac)	...	c_2
s_4	a	a	c	c	b	c	...	c_1
s_5	(abc)	a	b	d	e	d	...	c_1

A sequence s_i collects a list of ordered objects e_n, $s_i = \{e_1, e_2, ..., e_n\}$, in which $e_n = (x_1 x_2 ... x_q)$ is an element consisting of activities, events or actions in the behavior sequence, and x_q records the properties or items associated with the sequence itemset. When timestamps $(t_1, ..., t_n)$ are added to their corresponding behavior actions $(e_1, e_2, ..., e_n)$, we generate temporal sequences. A temporal sequence is expressed as $s_i = \{(t_1, e_1), (t_2, e_2), ..., (t_n, e_n)\}$ where $t_{(n-1)} < t_n$. In the real world, a sequence of behavior often incurs certain impact on business, for instance, a series of abnormal online payments incur online payment fraud, a list of high risk terrorist activities may lead to an eventual disaster to the society. Let $C = c_1, c_2, ..., c_m$ represent such business impacts, c_m is a specific class of impact, for instance, high risk customers. Table 1 shows an example of five sequences, each sequence consists of a list of actions happening at different time points. At some time point, multiple actions co-occur, such as $(t_1, s_5) = (abc)$. Each sequence is associated with a business impact label, for instance, s_5 has associated label c_1. In practice, quantitative temporal information associated with activities is helpful for distinguishing high impact behavior from others. We call such activities *time sensitive*. Time-sensitive behavior is widely seen in many applications. For instance,

- Example 1. In a medical diagnosis and symptom analysis, the temporal information is crucial for doctors to accurately diagnose diseases. For instance, H1N1 influenza (Swine flu) has a rapid onset within 3-6 hours, presenting with high fever (greater than 102 ^{o}F). In contrast, such sudden fever is rare with a common cold. This example shows the importance of considering temporal intervals in sequence analysis.
- Example 2. As for failure detection and identification in assembly line systems, anomaly can be detected with the help of the quantitative temporal intervals between tasks. For example, suppose there are three successive workflow tasks. It is 8 minutes from task 1 to task 2, and 2 minutes from task 2 to task 3. If a record shows 2 minutes from task 1 to task 2 and 6 minutes from task 2 to task 3, apparently this may indicate the presence of anomaly even though the sequence representation of those tasks present nothing abnormal. This example shows that sequence analysis without considering temporal intervals may miss important findings.
- Example 3. In the web usage analysis, if many users tend to stay for a longer time with some particular websites than visiting others, the browsing duration difference indicates more attractive value of the long-stay websites. This example shows the importance of considering user navigation duration in web usage analysis.

To analyze patterns in the above dataset in Table 1 and applications, traditional sequence analysis methods only count the ordering information among sequential items, and treat all actions equally by merging them together. For instance, a health insurant claims one to multiple service types at the same time with increasing frequencies may indicate either increasingly terrible health situation or fraudulent claims. Health insurance providers may be interested in claim review and active customer care, so as to work out why multiple services were conducted at the same time, whether there is any service of the patient's particular interest, why the patient frequently visited doctors, or whether the patient saw different doctors. While these questions are so critical for health insurance providers, it is hard for the existing sequence analysis approaches to find informative hints for these questions.

This is because the existing sequence analysis approaches mainly focus on sequence items, ordering relationship. Consequently, important information in temporal sequence is missing, for instance, the time interval between two consecutive activities, those co-occurring activities at the same time, and the impact label associated with a sequence. However, these aspects are critical for us to disclose in-depth causes and effects associated with discriminative behavior. For this, both temporal sequence analysis and temporal sequence classification can play an important role. Temporal sequence analysis is an emerging research issue in sequence analysis. Limited research has been conducted on mining sequential patterns from temporal sequential data. To the best of our knowledge, current approaches mainly pay attention to the timestamps associated with events, which are converted into sequential orders of the underlying activities.

In addition, while sequence classification is attracting more and more interest [3], people focus on the combination of classification with traditional sequential pattern mining. The goal of sequence classification is to predict which class a given sequence belonged to. No substantial work has been found on classifying temporal sequences.

Unfortunately, how to handle time sensitivity in the temporal sequence classification is a difficult problem. The construction of the sequence of items should be intertwined with the construction of its timestamps. Historically, researchers independently focus on either sequential or temporal aspects. How to combine the temporal information with sequence classification to attain an enhanced informative model is nearly unexploited. In addition, it is very time consuming to identify patterns combining temporal information with sequence classification.

In this paper, we discuss temporal sequence classification. The main idea is to incorporate temporal information into sequence classification. For this, we propose Temporal Feature Extraction Method (TFEM) to mine temporal features for sequence classification. Our contribution is two-fold.

- One is that we design innovative feature mining algorithms which can effectively represent temporal information for sequences classification. The time-sensitive features enrich temporal characteristics derived from the raw data, yielding improvement on sequence classification performance, as demonstrated by a set of experiments performed on synthetic and real-world datasets.
- The other is, our result is easily interpretable. We employ decision tree to generate human-friendly rules. Additionally, it provides an adaptive solution allowing user to determine a tradeoff between classification accuracy and computational cost.

The rest of the paper is organized as follows. Section 2 summarizes related work. Section 3 introduces our novel TFEM approach of mining time-sensitive features for sequence classification. Section 4 presents two empirical studies in which we applied our method to synthetic and real-world datasets. Section 5 discusses an extension of our TFEM approach. Finally we conclude our work in section 6.

2 Related Work

Temporal sequence mining has been explored intensively. Based on the nature of items, sequences can be divided into two categories: symbolic representation (discrete variable e.g. an action code, or tick-by-tick data) and time-series representation (continuous variable e.g. price in the stock market). Here we focus on symbolics as there are multiple approaches to covert time-series data into symbolics: for instance, Discrete Fourier transform (DFT) [4],Singular Value Decomosition(SVD) [5],Adaptive Piecewise constant approximation [6],Symblic Aggregate Approximation(SAX) [7].

There are enormous renowned classification algorithms. However, they are difficult to apply to sequential data, because there could be huge features potentially and thus intractable for relatively limited computing resources. In a seminal paper, Lesh etc. [8] proposed *FeatureMine* for sequence classification by analysing the presence of features derived from discriminative frequent patterns. The three phases of Lesh's method are:

1. Mining features. First of all, it adapts SPADE [9] to generate frequent patterns from sequence data. Chi-square tests are used to prune patterns to enforce discriminative and redundancy constraints. Remaining patterns $f_1, f_2, ..., f_n$ are outputted as features for classification.
2. Applying features to sequences. Most standard classifiers only accept an example as input when it is in the form of a vector consisting of feature-value pairs. Each feature generates a boolean value depending on its presence in a sequence. For example, if sequence s_i is "in presence of" pattern f_1 (i.e., f_1 is an subsequence of s_i), the value with regard to feature f_1 is true, otherwise it is false.
3. Classification. Based on the boolean feature-value pairs, traditional attribute-based classifiers can be used, such as Winnow and Naive Bayes.

After that, [10,11,12] incorporate biological knowledge into DNA sequence classification. Recently, there are overwhelming tools on protein sequences [13,14,15]. [16] uses implicit motif distribution based hybrid computational kernel for sequence classification. But to our best knowledge, combining sequence classification with temporal information is nearly unexploited.

3 A Novel TFEM Approach

As discussed in previous section, most existing sequence classification approaches seldom explicitly take time intervals between items into consideration. To address this limitation, we propose temporal feature extraction method (TFEM) to capture the interval characteristic.

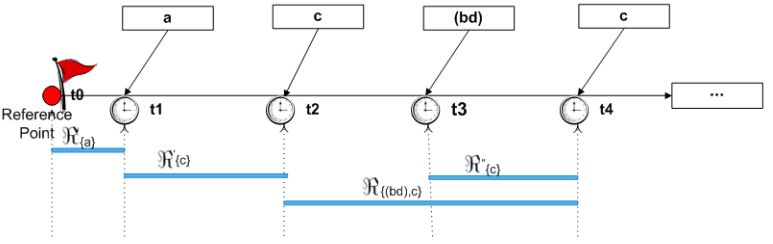

Fig. 1. A timeline representation of partial sequence s_1 from t_1 to t_4

Definition 1. *A behavior sequence* $s_i = \{e_1, e_2, ..., e_n\}$*, in which* $e_n = (x_1 x_2 ... x_q)$ *is an atomic item consisting of activities, events or actions in the behavior sequence, and* x_q *records the properties or items associated with the sequence itemset. If* $q = 1$*,* e_n *is a* ***single atomic item****, otherwise it is* ***composite atomic item****.*

Definition 2. *For an atomic item* e_n*,* $t_c[e_n], t_a[e_n]$ *denote the time stamps of current item and previous item a sequence* s_i*, respectively. In particular, for the first item in* s_i*,* $t_a = t_0$*, which is a* ***reference time*** *or start point for calculation.*

Definition 3. *If a pattern p contains only one atomic item, p is called 1-itemset; otherwise we name the first item in p as* $p_{firstItem}$ *and the last item as* $p_{lastItem}$*. An interval* $\Re$ *for pattern p in sequence* s_i *is defined as*

$$\Re = \begin{cases} Avg(t_c[p] - t_a[p]), & \text{p is 1-itemset,} \quad (1) \\ Avg(t_c[p_{lastItem}] - t_a[p_{firstItem}]), & \text{Otherwise.} \quad (1') \end{cases}$$

If pattern p repeats in s_1, an average value is taken when calculating $\Re$.

An example of calculating intervals within s_1 from t_1 to t_4 is depicted in Fig. 1. For instance, for 1-itemset {a},$\Re_{\{a\}} = t_1 - t_0$. For 2-itemset {(bd),c},$p_{firstItem}$ is {(bd)} and $p_{lastItem}$ is{c}. Therefore, $v_{\{(bd),c\}} = t_4 - t_2$. Again for 1-itemset pattern {c},it occurs twice in s_1. For the first presence of {c}, the interval $\Re'_{\{c\}} = t_2 - t_1$ and for the second presence, $\Re''_{\{c\}} = t_4 - t_3$. $\Re_{\{c\}} = (\Re'_{\{c\}} + \Re''_{\{c\}})/2 = (t_4 - t_3 + t_2 - t_1)/2$.

The basic idea of our TFEM approach is during the traditional feature extraction for sequence classification, we incorporate interval information to create more informative features and thus classifier can take advantage of those constructed new TFEM features.

3.1 Framework

The dataflow of our TFEM sequence classification is described in Figure 2. The whole process is divided into three phrases:

– **Data Representation and Preprocessing:** First of all, sequential pattern mining algorithm is employed to get initial features (Basically they are frequent patterns extracted from raw data and have been pruned by statistical tests). Then we calculate an interval for each pattern in each sequence. Thus we can generate 2-tuple (pattern, interval) pairs for every sequence.

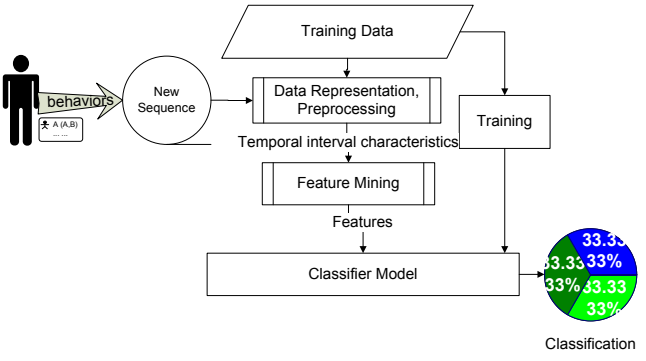

Fig. 2. A dataflow of behavior sequence classification

- **Feature Mining:** The TFEM algorithm in section 3.3 is designed to construct new temporal features for sequence classification.
- **Training and Testing:** 10 fold cross-validation is conducted. Decision tree classifier is used to generate easily interpretable rules. Then the trained classifier makes predictions on incoming sequences.

3.2 Data Representation and Preprocessing

By using featureMine proposed by Lesh etc. [8], we attain patterns $\{a\}, \{(ac)\}, \{b\}, \{a, b\}, ...$ as our initial features in the previous example. Then for each pattern f_i in every sequence we calculate its interval using formula 1 and generate 2-tuple (pattern, interval) pair, which is shown in table 2.

Table 2. An example dataset in (pattern, interval) pairs

ID	$\{a\}$	$\{b\}$	$\{a, b\}$	$\{(ac)\}$	...
s_1	1	1	2	1	...
s_2	1	1	3		...
s_3	1	1	3	3	...
s_4	1	1	2		...
s_5	1	1	2	1	...

3.3 Feature Mining

Construction of Temporal Features. We design TFEM temporal feature algorithm to construct new temporal features, which is described in Fig. 3.

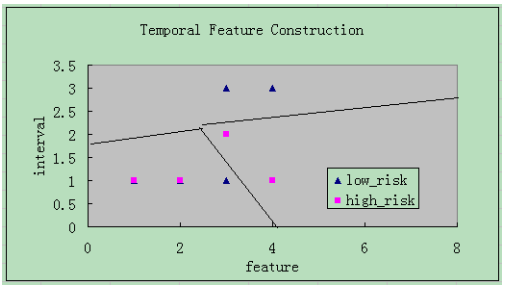

Fig. 3. A example of temporal feature construction

Algorithm 1: Temporal Feature Extraction Algorithm

Input: Min_freq(c_i),Dataset D.
Output: Candidate temporal features.

(a). Represent data as 2-tuple (pattern, interval) pairs in the two dimensional feature space.
(b). Merge and cluster. In order to reduce the feature space, for the same pattern p, intervals are merged if they belong to the same class. For example, if any feature examples generated in previous step from (pattern 1, 8) to (pattern 1, 10) are all positive, they can be merged as (pattern 1, 8∼10). For those belonging to multiple classes, we adopt an odds-ratio test and simply prune points which are less skewed in the class distribution. For example, in two-classes classification, we calculate the discriminative power by the following formula:

$$E = \frac{p_1/(1-p_1)}{p_2/(1-p_2)} \tag{2}$$

where p_1, p_2 are proportions of a pattern in difference classes respectively. Divide our (pattern,interval) space into several regions by clustering. It is shown that there are three regions in Fig. 3.
(c). Output region boundaries as candidate temporal features. In our example, the three regions are our newly constructed temporal features.

The next step is to make use of these regions. For an incoming sequence, we check every pattern's presence. If the pattern occurs then calculate its interval value and locate its point in (pattern, interval) two-dimension feature space. The temporal feature value is true or false depending on which region it falls in.

Temporal Feature Selection. After constructing new temporal features, statistical optimization is performed in order to achieve highly efficient classification. There are three pruning criteria in our algorithm:

1. Features should be frequent and with strong discriminative power.
2. Features should be efficient for classification.
3. Features should be optimized, without complex parameter tuning.

This process is described in algorithm 2.

Algorithm 2: Temporal Feature Mining Algorithm

Input:Dataset D in the form of (pattern,interval) pair.
Output: Temporal features.

(a). Generate candidate features by previous feature extraction algorithm.
(b). Prune any candidate if it meets any criterion in the following tests:
- Discriminative test: The odds-ratio test is employed to ensure features are significantly discriminative among classes.
- Redundancy test: We create new calculation formula based on Foil-Gain [17] to estimate information gain. For instance, regarding to biclassification

$$E = Max(tw(\log_2 \frac{p_1}{p_1 + n_1} - \log_2 \frac{p_2}{p_2 + n_2})) \quad (3)$$

where p_1, n_1 is that number of positive and negative examples covered before adding new feature. p_2, n_2 is that number of positive and negative examples covered when adding one new feature. t is the number of positive examples covered by both. w is the proportion of pattern's duration time in global temporal dimension.
- Optimization test: We tune our model by enumerate parameters' thresholds. For instance, the threshold for pattern's length can be determined by simply the trial and error method, that is, running our tests with different length and selecting the best.

(c). Output newly constructed features after pruning in step b.

3.4 Training and Testing

We choose a rule-based classification method for several reasons. First, it generates human-readable results. This is very important for the interpretability of our model in practice. Secondly, it is efficient. The time complexity is O(N) while N is the number of rules. Finally, with respect to imbalance data, rule-based learner is more effective.

Based on our temporal features, classifier can improve its accuracy as those constructed features help to capture informative temporal characteristics in the raw data.

4 Empirical Studies

In order to evaluate our methods, we implement TFEM in both symbolic sequences and time-series datasets.

4.1 Health Insurance Dataset

We use a health insurance dataset to test our TFEM framework, which describes every member's (or user's) claim history. In our experiment, there are a total of 15875 records from 479 users. Each record is in the format of 4-tuple vector (member_id, service_date, service_code, server_content). We reorganize the data into sequences based on the attribute of *member_id* in a temporal order. This dataset contains a sample of

Table 3. Traditional sequence classification confusion matrix

accuracy: 76.41 %			
	true high-risk	true low-risk	class precision
pred. high-risk	198	71	73.61%
pred. low-risk	42	168	80.00 %
class recall	82.50 %	70.29%	

Table 4. TFEM sequence classification confusion matrix

accuracy: 83.11 %			
	true high-risk	true low-risk	class precision
pred. high-risk	183	24	88.41 %
pred. low-risk	57	215	79.04%
class recall	76.25 %	89.96%	

479 sequences with unequal length. Each sequence depicts a member's claim history. Besides, each sequence in the training set has been labeled as either "high-risk" or "low-risk". Table 1 shows a sample of our dataset. For privacy preserving, a, b, c denote the abstraction of actions in each real-world sequence record. c_1 represents high-risk class label while c_2 is low-risk class label. Apparently, two items may happen in the same time. For example, in sequence s_1, a and c are both associated with time-stamp t_6.

Our algorithms are developed by Java 1.6, under Eclipse 3.2 environments. Hardware of our computer is duo-core Intel Pentium 4.2 with 1.5 G memory.

We conduct sequence classification on the insurance data. After frequency pattern mining phrase, we obtain 80 features with min_support=48. The art for choosing an appropriate min_support threshold is to make sure our feature set is neither too big nor too small. In this discriminative test, the parameter value of odd-rate is 2. We use 10-fold cross validation and calculate classification accuracy. Table 3 describes the performance of Lesh's method as a benchmark. By comparison, table 4 shows the performance of TFEM model. From the performance contrast test, we can see the TFEM framework can increase the accuracy from 76.41% to 83.11%.

4.2 Ionosphere Dataset

The ionosphere dataset is downloaded from UCI KDD repository [18]. The time-series data was collected by a system in Goose bay, Labrador. There are two classes in a total of 351 samples. After converting those time series data, we run traditional frequent pattern based sequence classification and our TFEM approach. The result shows TFEM outperforms its conventional counterpart with an increase in accuracy from 76.13% to 81.09%.

4.3 Effects of Varying Odds-Ratio

Fig. 4 shows comparison of traditional method and TFEM under several odds-ratio parameter settings. We adjust different odds-ratio and measure the accuracy and time-cost.

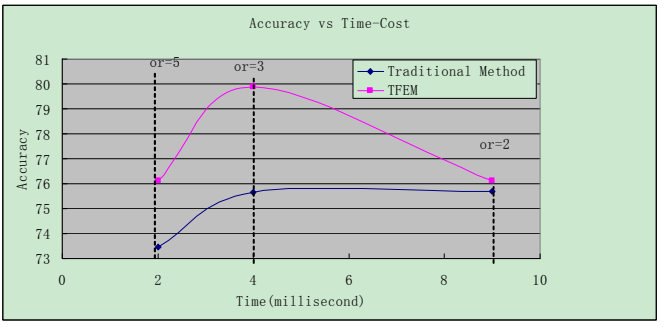

Fig. 4. Accuracy vs. time-cost

It is observed that the greater the value of odds-ratio parameter is, the more candidate features pruned, which reduces overall time-cost. On the other hand, higher accuracy will lead to longer feature extraction time. Flexibility is offered with a tradeoff between classification accuracy and time-cost.

5 Discussion

In this section, we first employ PCA [19] to reduce the computation cost in our algorithms and make TFEM more efficient. Then we discuss about handling time-series data.

Principal component analysis (PCA) describes a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal component. PCA was first invented in 1901 by Karl Pearson [20]. PCA [21,22] is mathematically an orthogonal linear transformation that transforms data to a new coordinate system. As you can see from our insurance experiment, there are 23 features. In some cases, in order to find better fine granularity for frequent patterns, we may end up with hundreds of features. Therefore, PCA is used to optimize our model. Fig. 5 depicts the cumulative proportion of variance. In this way, the number

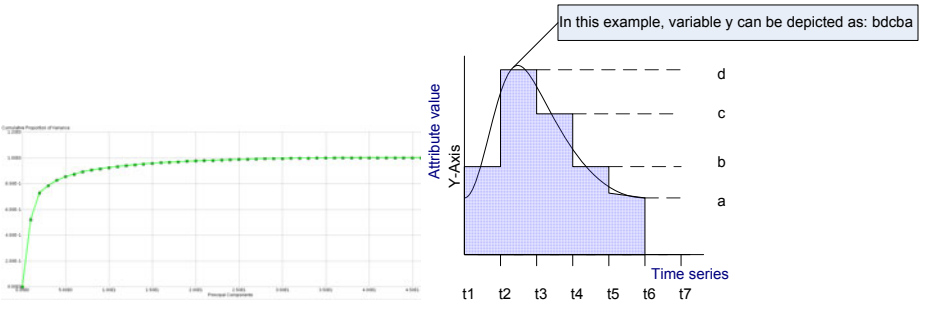

Fig. 5. Principal components analysis and shift to symbolic events

of features can be significantly reduced and only the most representative instances are kept.

Fig. 5 also shows how to convert continuous variables into the symbolic representation. This method is based on Yi and Faloutsos and Keogh et al.'s Piecewise Aggregate Approximation (PAA) [23]. In PAA, each record of time series data is divided into k segments with equal length and the average value of each segment is used as data-reduced representation. Obviously the PAA model is very straightforward and easy to implemented. It is very fast and has almost linear time complexity. But on the other hand, it may lose useful information and a variable indicating the slope in each segment becomes useful during the conversion process.

6 Conclusion

Quantitative temporal information associated with activities is helpful for distinguishing high impact behavior from others in many business problem-solving. In this paper, we proposed a novel temporal feature extraction for behavior sequence classification. TFEM incorporates time intervals, which are critical in many business applications, into behavior sequence classification. With informative features, experiments show the performance of classifier is significantly improved.

TFEM is of great significance for discovering knowledge from time-sensitive behavior sequences. Furthermore, it is important to note that TFEM can be easily extended to handle other characteristics without being limited to temporal dimension, such as spatial space.

Acknowledgments. This work is sponsored in part by Australian Research Council Discovery Grants (DP1096218, DP0988016, DP0773412) and ARC Linkage Grant (LP0989721, LP0775041).

References

1. Foxall, C., James, V.: Behavior Analysis of Consumer Brand Choice: A Preliminary Analysis1. The Behavioral Economics of Brand Choice, p. 54 (2007)
2. Cao, L.: Behavior informatics and analytics: Let behavior talk. In: ICDM Workshops, pp. 87–96. IEEE Computer Society, Los Alamitos (2008)
3. Lesh, N., Zaki, M.J., Ogihara, M.: Mining features for sequence classification. In: KDD 1999: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 342–346. ACM, New York (1999)
4. Brigham, E., Yuen, C.: The fast Fourier transform. IEEE Transactions on Systems, Man and Cybernetics 8(2), 146–146 (1978)
5. Golub, G., Reinsch, C.: Singular value decomposition and least squares solutions. Numerische Mathematik 14(5), 403–420 (1970)
6. Eriksson, K., Estep, D., Hansbo, P., Johnson, C.: Introduction to adaptive methods for differential equations. Acta numerica 4, 105–158 (2008)
7. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing SAX: a novel symbolic representation of time series. Data Mining and Knowledge Discovery 15(2), 107–144 (2007)

8. Lesh, N., Zaki, M., Ogihara, M.: Mining features for sequence classification. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 342–346. ACM, New York (1999)
9. Zaki, M.: SPADE: An efficient algorithm for mining frequent sequences. Machine Learning 42(1), 31–60 (2001)
10. Ma, Q., Wang, J., Shasha, D., Wu, C.: DNA sequence classification via an expectation maximization algorithm and neural networks: a case study. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 31(4), 468–475 (2001)
11. Rätsch, G., Sonnenburg, S., Schäfer, C.: Learning interpretable SVMs for biological sequence classification. BMC bioinformatics 7(Suppl. 1), S9 (2006)
12. Ferreira, P., Azevedo, P.: Protein sequence classification through relevant sequence mining and bayes classifiers. In: Bento, C., Cardoso, A., Dias, G. (eds.) EPIA 2005. LNCS (LNAI), vol. 3808, pp. 236–247. Springer, Heidelberg (2005)
13. Mulder, N., Apweiler, R.: InterPro and InterProScan: tools for protein sequence classification and comparison. Methods in Molecular Biology (Clifton, NJ) 396, 59 (2007)
14. Shen, L., Satta, G., Joshi, A.: Guided learning for bidirectional sequence classification. In: Annual Meeting-Association for Computational Linguistics, vol. 45, p. 760 (2007)
15. Spurdle, A., Lakhani, S., Healey, S., Parry, S., Da Silva, L., Brinkworth, R., Hopper, J., Brown, M., Babikyan, D., Chenevix-Trench, G., et al.: Clinical classification of BRCA1 and BRCA2 DNA sequence variants: the value of cytokeratin profiles and evolutionary analysis–a report from the kConFab Investigators. Journal of Clinical Oncology 26(10), 1657 (2008)
16. Atalay, V., Cetin-Atalay, R.: Implicit motif distribution based hybrid computational kernel for sequence classification. Bioinformatics 21(8), 1429–1436 (2005)
17. Quinlan, J.: Learning logical definitions from relations. Machine learning 5(3), 239–266 (1990)
18. Uci kdd repository, `http://archive.ics.uci.edu/ml/datasets/Ionosphere:`
19. Jolliffe, I.: Principal component analysis. Springer, Heidelberg (2002)
20. Gorban, A., Kgl, B., Wunsch, D., Zinovyev, A.: Principal manifolds for data visualization and dimension reduction, p. 340. Springer Publishing Company, Heidelberg (2007) (incorporated)
21. Rohlf, F.: Morphometric spaces, shape components and the effects of linear transformations. In: Advances in morphometrics, pp. 117–129 (1996)
22. Cai, D., He, X., Han, J., Zhang, H.: Orthogonal laplacianfaces for face recognition. IEEE Transactions on Image Processing 15(11), 3608–3614 (2006)
23. Keogh, E., Pazzani, M.: Scaling up dynamic time warping for datamining applications. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 285–289. ACM, New York (2000)