A Brief Survey on Sequence Classification

Zhengzheng Xing
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
zxing@cs.sfu.ca

Jian Pei
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
jpei@cs.sfu.ca

Eamonn Keogh
Department and Computer Science and Engineering
University of California, Riverside, CA, USA
eamonn@cs.ucr.edu

ABSTRACT

Sequence classification has a broad range of applications such as genomic analysis, information retrieval, health informatics, finance, and abnormal detection. Different from the classification task on feature vectors, sequences do not have explicit features. Even with sophisticated feature selection techniques, the dimensionality of potential features may still be very high and the sequential nature of features is difficult to capture. This makes sequence classification a more challenging task than classification on feature vectors. In this paper, we present a brief review of the existing work on sequence classification. We summarize the sequence classification in terms of methodologies and application domains. We also provide a review on several extensions of the sequence classification problem, such as early classification on sequences and semi-supervised learning on sequences.

1. INTRODUCTION

Sequence classification has a broad range of real-world applications. In genomic research, classifying protein sequences into existing categories is used to learn the functions of a new protein [13]. In health-informatics, classifying ECG time series (the time series of heart rates) tells if the data comes from a healthy person or comes from a patient with heart disease [59]. In anomaly detection/intrusion detection, the sequence of a user's system access activities on Unix is monitored to detect abnormal behaviors [33]. In information retrieval, classifying documents into different topic categories has attracted a lot of attentions [51]. Other interesting examples include classifying query log sequences to distinguish web-robots from human users [58; 18] and classifying transaction sequence data in a bank for the purpose of combating money laundering [42].

Generally, a sequence is an ordered list of events. An event can be represented as a symbolic value, a numerical real value, a vector of real values or a complex data type. In this paper, we consider sequence data into the following subtypes.

- Given an alphabet of symbols $\{E_1, E_2, E_3, ..., E_n\}$, a *simple symbolic sequence* is an ordered list of the symbols from the alphabet. For example, a DNA sequence is composed of four animo acid A, C, G, T and a DNA segment, such as $ACCCCCGT$, is a simple symbolic sequence.
- A *complex symbolic sequence* is an ordered list of vectors. Each vector is a subset of the alphabet [34]. For example, for a sequence of items bought by a customer over one year, treating each transaction as a vector, a sequence can be $\langle (milk, bread)(milk, egg) \cdots (potatos, cheese, coke) \rangle$.
- A *simple time series* is a sequence of real values ordered in timestamp ascending order. For example,

  $$\langle (t_1, 0.1)(t_2, 0.3) \cdots (t_n, 0.3) \rangle$$

  is a simple time series recording the data from time stamp t_1 to t_n.
- A *multivariate time series* is a sequence of numerical vectors. For example,

  $$\langle (t_1, \langle 0.1, 0.3, 05 \rangle)(t_2, \langle 0.3, 0.9, 0.8 \rangle) \cdots (t_n, \langle 0.3, 0.9, 0.4 \rangle) \rangle$$

  is a multivariate time series.
- In the above, the data types of the events are simple. In some applications, the data type of events can be arbitrarily complicated. For example, in a patient record data set (`http://www.informsdmcontest2009.org/`), each patient is represented by a longitudinal sequence of hospital visits. Each visit is an event and is described by multiple numerical measurements, categorical fields and text descriptions. A *complex event sequence* refers to the general form of sequences.

A sequence may carry a class label. For example, a time series of ECG data may come from a healthy or ill person. A DNA sequence may belong to a gene coding area or a non-coding area. Given L as a set class labels, the task of *(conventional) sequence classification* is to learn a *sequence classifier* C, which is a function mapping a sequence s to a class label $l \in L$, written as, $C : s \rightarrow l, l \in L$.

In (conventional) sequence classification, each sequence is associated with only one class label and the whole sequence is available to a classifier before the classification. There are also other application scenarios for sequence classification. For example, for a sequence of symptoms of a patient over a long period of time, the health condition of the patient may change. For a streaming sequence, which can be regarded as a virtually unlimited sequence, instead of predicting one class label, it is more desirable to predict a sequence of labels. This problem is considered in [24; 23] as the *strong sequence classification* task. In this paper, we will discuss several extensions of (conventional) sequence classification in Section 3.

There are three major challenges in sequence classification. First, most of the classifiers, such as decision trees and neural networks, can only take input data as a vector of features. However, there are no explicit features in sequence data. Second, even with various feature selection methods, we can transform a sequence into a set of features, the feature selection is far from trivial. The dimensionality of the feature space for the sequence data can be very high and the computation can be costly. Third, besides accurate classification results, in some applications, we may also want to get an interpretable classifier. Building an interpretable sequence classifier is difficult since there are no explicit features.

In this paper, we give a brief survey of the existing sequence classification methods. Since most of the existing works focus on the task of conventional sequence classification, Section 2 is devoted to summarizing the major methods for this task. In Section 3, we discuss some extensions of the conventional sequence classification tasks, such as streaming sequence classification and early classification on sequences. In Section 4, we summarize sequence classification from the perspective of application domains, such as time series data, text data and genomic data. Section 5 concludes the paper.

2. SEQUENCE CLASSIFICATION METHODS

The sequence classification methods can be divided into three large categories.

- The first category is feature based classification, which transforms a sequence into a feature vector and then apply conventional classification methods. Feature selection plays an important role in this kind of methods.
- The second category is sequence distance based classification. The distance function which measures the similarity between sequences determines the quality of the classification significantly.
- The third category is model based classification, such as using hidden markov model (HMM) and other statistical models to classify sequences.

In the rest of this section, we will present some representative methods in the three categories. Some methods may ride on multiple categories. For example, we can use SVM by either extracting features (Category 1) or defining a distance measure (Category 2). Sequence classification using SVM will be summarized in Section 2.3. All methods discussed in this section are for conventional sequence classification.

2.1 Feature Based Classification

Conventional classification methods, such as decision trees and neural networks, are designed for classifying feature vectors. One way to solve the problem of sequence classification is to transform a sequence into a vector of features through feature selections.

For a symbolic sequence, the simplest way is to treat each element as a feature. For example, a sequence $CACG$ can be transformed as a vector $\langle A, C, C, G\rangle$. However, the sequential nature of sequences cannot be captured by this transformation. To keep the order of the elements in a sequence, a short sequence segment of k consecutive symbols, called a k-gram, is usually selected as a feature. Given a set of k-grams, a sequence can be represented as a vector of the presence and the absence of the k-grams or as a vector of the frequencies of the k-grams. Sometimes, we also allow inexact matchings with gapped k-grams. By using k-grams as features, sequences can be classified by a conventional classification method, such as SVM [35; 36] and decision trees [12]. A summary of k-gram based feature selection methods for sequence classifications can be found in [16].

The size of candidate features which are all k-grams where $1 \leq k \leq l$ is $2^l - 1$. If k is a large number, the size of the feature set can be huge. Since not all features are equally useful for classification, Chuzhanova *et al.* [12] use Gamma test to select a small informative subset of features from the k-grams. A genetic algorithm is used to find the local optimal subset of features.

In contrast to k-gram based feature selections, Lesh *et al.* [30; 34] propose a pattern-based feature selection method. The features are short sequence segments which satisfy the following criteria (1) frequent in at least one class (2) distinctive in at least one class and (3) not redundant. Criterion (2) means a feature should be significantly correlated with at least one class. The redundancy in Criterion (3) can be defined in the way of feature specification and feature generalization. An efficient feature mining algorithm is proposed to mine features according to the criteria. After selecting the features, Winnow [41] and naive bayes classifiers are used. The experimental results in [30] show that comparing to the method of considering each element as a feature, pattern-based feature selection can improve the accuracy by 10% to 15%.

The challenge of applying pattern-based feature selection on symbolic sequences is how to efficiently search for the features satisfying the criteria. Ji *et al.* [22] propose an algorithm to mine distinctive subsequences with a maximal gap constraint. The algorithm, which uses bisect and boolean operations and a prefix growth framework, is efficient even with a low frequency threshold.

Time series data is numeric. The feature selection techniques for symbolic sequences cannot be easily applied to time series data without discretization. Discretization may cause information lost. Ye *et al.* [65] propose a feature selection method which can be applied directly on numeric time series. Time series shapelets, the time series subsequences which can maximally represent a class, is proposed as the features for time series classification. For a two-class classification task, given a distance threshold, a shapelet is a segment of time series which can be used to separate the training data into two parts according to the distance to the shapelet, and maximizes the information gain. The distance threshold and the shapelet are learned from the training data to optimize the information gain. To construct a classifier, the shapelet selection process is integrated with the construction of the decision tree.

Although subsequences are informative features, they can only describe the local properties of a long sequence. Aggarwal *et al.* [5] develop a method to capture both the global and local properties of sequences for the purpose of classification. Aggarwal *et al.* [5] modify wavelet decomposition to describe a symbolic sequence on multiple resolutions. With different decomposition coefficients, the wavelet represents the trends in different range of intervals, from global to local. Using wavelet decomposition and a rule based classi-

fier, the wavelet decomposition method outperforms the k-nearest neighbor classifier on a web accessing sequence data set and on a genomic sequence data set.
In summary, the existing methods differ from each other on the following aspects.

- Which criteria should be used for selecting features, such are distinctiveness, frequency, and length?
- In which scope does feature selection reflect the sequential nature of a sequence, local or global?
- Should matchings be exact or inexact with gaps?
- Should feature selection be integrated within the process of constructing the classifier or a separate preprocessing step?

2.2 Sequence Distance Based Classification

Sequence distance based methods define a distance function to measure the similarity between a pair of sequences. Once such a distance function is obtained, we can use some existing classification methods, such as K nearest neighbor classifier (KNN) and SVM with local alignment kernel (to be discussed in Section 2.3 [49], for sequence classification.
KNN is a lazy learning method and does not pre-compute a classification model. Given a labeled sequence data set T, a positive integer k, and a new sequence s to be classified, the KNN classifier finds the k nearest neighbors of s in T, $kNN(s)$, and returns the dominating class label in $kNN(s)$ as the label of s.
The choice of distance measures is critical to the performance of KNN classifiers. In the rest of this section, we focus on summarizing different distance measures proposed for sequence data.
For simple time series classification, Euclidean distance is a widely adopted option [26; 59]. For two time series s and s', Euclidean distance is

$$dist(s, s') = \sqrt{\sum_{i=1}^{L} (s[i] - s'[i])^2}.$$

The Euclidean distance usually requires two time series to have the same length. Keogh *et al.* [26] show when applying 1NN classifier on time series, Euclidean distance is surprisingly competitive in terms of accuracy, compared to other more complex similarity measures.
Euclidean distance is sensitive to distortions in time dimension. Dynamic time warping distance (DTW) [28] is proposed to overcome this problem and does not require two time series to be of the same length. The idea of DTW is to align two time series and get the best distance by aligning. One example of DTW is shown in Figure 1. Xi *et al.* [61] show that on small data sets, elastic measures such as dynamic time warping (DTW) can be more accurate than Euclidean distance. However, recent empirical results [15] strongly suggest that on large data sets, the accuracy of elastic measures converges with Euclidean distance.
Dynamic time warping is usually computed by dynamic programming and has the quadratic time complexity. Therefore, it is costly on a large data set. Ratanamahatana *et al.* [48] propose a method to dramatically speed up the DTW similarity search process by using tight lower bounds

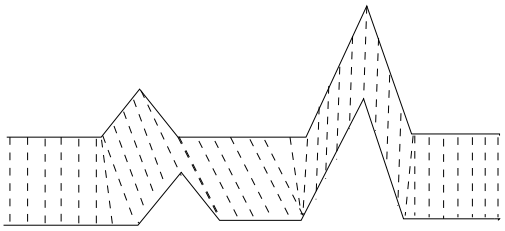

Figure 1: Dynamic Time Warping

to prune may calculations. Xi *et al.* [61] use numerical reduction to speed up DTW computation. The idea is to reduce the number of the training examples used by a 1NN classifier and, at the same time, adjust the warping window dynamically.
For symbolic sequences, such as protein sequences and DNA sequences, alignment based distances are popular adopted [25]. Given a similarity matrix and a gap penalty, the Needleman-Wunsch algorithm [44] computes an optimum global alignment score between two sequences through dynamic programming. In contrast to global alignment algorithms, local alignment algorithms, such as the Smith-Waterman algorithm [53] and BLAST [6], measure the similarity between two sequences by considering the most similar regions but not enforcing the alignments on full length.

2.3 Support Vector Machine

SVM has been proved to be an effective method for sequence classification [43; 39; 35; 54; 55; 52; 13]. The basic idea of applying SVM on sequence data is to map a sequence into a feature space and find the maximum-margin hyperplane to separate two classes. Sometimes, we do not need to explicitly conduct feature selection. A kernel function corresponds to a high dimension feature space. Given two sequences, x, y, some kernel functions, $K(x, y)$, can be viewed as the similarity between two sequences [54]. The challenges of applying SVM to sequence classification include how to define feature spaces or kernel functions, and how to speed up the computation of kernel matrixes.
One of the widely used kernels for sequence classification is *k-spectrum kernel* or *string kernel*, which transforms a sequence into a feature vector. Leslie *et al.* [35] propose a k-spectrum kernel for protein classification. Given the protein animo acid alphabet of 20 elements $\langle A, R, N, D \cdots \rangle$, the k-spectrum is all the possible sequences of length k that are composed by the elements in the alphabet. For example, if $k = 3$, the k-spectrum contains ARN, AND, DCN, and so on. Given the alphabet $\mathcal{A}$, a sequence x is transformed into a feature space by a transformation function

$$\Phi_k(x) = (\phi_a(x))_{a \in \mathcal{A}^k}$$

where $\phi_a(x)$ is the number of times a occurs in x. The kernel function is the dot product of the feature vectors,

$$K(x, y) = \Phi_k(x) \cdot \Phi_k(y)$$

By using a suffix tree algorithm [35], $K(x, y)$ can be computed in $O(kn)$ time.
Lodhi *et al.* [43] propose a string kernel for text classification. Similar to the k-spectrum kernel in [35], the string kernel also uses a k-length sub-sequences but allows gaps. By using an exponentially decaying factor of the length of

span of the subsequences occurring in the text, the gap is penalized. The kernel function is the dot product of the feature vectors and can be efficiently computed by dynamic programming. Leslie *et al.* [36] extends the k-spectrum kernel to handle mismatching. Sonnenburg *et al.* [55] propose a fast k-spectrum kernel with mismatching .

One disadvantage of kernel based methods is that it is hard to be interpreted and hard for users to gain knowledge besides a classification result. Sonnenburg *et al.* propose a method to learn interpretable SVMs using a set of a string kernels [54]. The ideas is to use a weighted linear combination of base kernels. Each base kernel uses a distinctive set of features. The weights represent the importance of the features. After learning the SVM, users can have an insight into the importance of different features.

String kernels or k-spectrum kernel can be viewed as a feature based method. Saigo *et al.* [49] propose a *local alignment kernel* for protein sequence classification which can be viewed as a distance based method. Although local alignment distance can effectively describe the similarity between two sequences, it cannot be directly used as a kernel function because it lacks the positive definiteness property. Saigo *et al.* [49] modify the local alignment distance and form a valid kernel called local alignment kernel, which mimics the behavior of the local alignment. The theoretical connection between the local alignment kernel and the local alignment distance is proved. Given two sequences x, y, the local alignment kernel $K(x, y)$ can be computed by dynamic programming.

Other kernels used for sequence classification include polynomial-like kernels [52], kernels derived from probabilistic model (Fisher's kernel) [52], and diffusion kernels [50].

2.4 Model Based Classification

One category of sequence classification methods is based on generative models, which assume sequences in a class are generated by an underlying model M. Given a class of sequences, M models the probability distribution of the sequences in the class. Usually, a model is defined based on some assumptions, and the probability distributions are described by a set of parameters. In the training step, the parameters of M are learned. In the classification step, a new sequence is assigned to the class with the highest likelihood.

The simplest generative model is the Naive Bayes sequence classifier [37]. It makes the assumption that, given a class, the features in the sequences are independent of each other. The conditional probabilities of the features in a class are learned in the training step. Due to its simplicity, Naive Bayes has been widely used from text classification [29] and genomic sequences classification [11].

However, the independence assumption required by Naive Bayes is often violated in practice. Markov Model and Hidden Markov Model can model the dependence among elements in sequences [17].

Yakhnenko *et al.* [64] apply a k-order Markov model to classify protein and text sequence data. In the training process, the model is trained in a discriminative setting instead of the conventional generative setting to increase the classification power of the generative model based methods.

Different from Markov Model, Hidden Markov Model assumes that the system being modeled is a Markov process with unobserved states. Srivastava *et al.* [56] use a profile HMM to classify biological sequences. A profile HMM usually has three types of states, inserting, matching and deleting. Aligned training examples are used to learn the transition probabilities between the states and emission probabilities. The learned HMM represents the profile of the training dataset. A profile HMM may also be learned from the unaligned sequences by gradually aligning each example with the existing profile. For each class, a profile HMM is learned. In the classification step, an unknown sequence is aligned with the profile HMM in each class by dynamic programming. An unknown sequence will be classified into the class which has the highest alignment score.

3. EXTENSIONS OF SEQUENCE CLASSIFICATION

In this section, we review some closely related or extended problems of conventional sequence classification. Those extensions are proposed to address the challenges when applying sequence classification to different real world application scenarios, such as classifying a sequence using its prefixes to achieve early classification, classifying sequences by using both labeled and unlabeled data, and predicting a sequence of labels instead of a single label for streaming sequences.

3.1 Early Classification

For temproal symbolic sequences and time series, the values of a sequence are received in time stamp ascending order. Sometimes, monitoring and classifying sequences as early as possible is desired. For example, in a retrospective study of the infants admitted to a neonatal intensive care unit, it is found that the infants had abnormal heart beating time series pattern 24 hour before the doctor finally diagnosed them with sepsis [21]. As another example, Bernaille *et al.* [9] show that by only observing the first five packages of a TCP connection, the application associated with the traffic flow can be classified. The applications of online traffic can be identified without waiting for the TCP flow to end. Generally, early classification of sequences may have applications in anomaly detection, intrusion detection, health informatics, and process control.

To the best of our knowledge, Diez *et al.* [14] first mentioned the concept of early classification of time series. They describe a time series by some relative literals, such as "increase" and "stay", and some region literals, such as "always" and "sometimes" over some intervals. Each literal and its associated position are viewed as a base classifier. Ada boost [19] is used to ensemble the base classifiers. The ensemble classifier is capable of making predictions on incomplete data by viewing unavailable suffixes of sequences as missing features.

Anibal *et al.* [8] apply a case based reasoning method to classify time series to monitor the system failure in a simulated dynamic system. The KNN classifier is used to classify incomplete time series using various distances, such as Euclidean distance and Dynamic time warping (DTW) distance. The simulation studies show that, by using case based reasoning, the most important increase of classification accuracy occurs on the prefixes through thirty to fifty percent of the full length.

Although in [14; 8], the importance of early classification on time series is identified and some encouraging results are shown, the study only treat early classification as a problem

of classifying prefixes of sequences. Xing *et al.* [62] point out the challenge of early classification is to study the tradeoff between the earliness and the accuracy of classification. The methods proposed in [14; 8] only focus on making predictions based on partial information but do not address how to select the shortest prefix to provide a reliable prediction. This makes the result of early classification cannot be easily used by users for further actions

Xing *et al.* [62] formulate the early classification problem as classifying sequences as early as possible while maintaining an expected accuracy. A feature based method is proposed for early classification on temporal symbolic sequences. The major idea is to first select a set of features that are frequent, distinctive and early, and then build an association rule classifier or a decision tree classifier using those features. In the classification step, an oncoming sequence is matched with all rules or branches simultaneously until on a prefix, a matching is found and the sequence is classified. In this way, a sequence is classified immediately once the user expected accuracy is achieved. The methods proposed in [62] show some successes in handling symbolic sequences by achieving competitive accuracies using only less than half of the length of the full sequences.

One disadvantage of the methods in [62] is that it cannot handle numeric time series well. Since numeric time series need to be discretized online, the information loss makes some distinctive features not easy to capture. Xing *et al.* [63] propose an early classifier for numeric time series by utilizing instance based learning. The method learns a minimal prediction length (MPL) for each time series in the training dataset through clustering and uses MPLs to guide early classification. As shown in Section 2, 1NN classifier with Euclidean distance is a highly accurate classifier for time series classification. One interesting property of the method in [63] is that without requiring a user expected accuracy, the classifier can achieve early classification while maintain roughly the same accuracy as a 1NN classifier using full length time series.

3.2 Semi-Supervised Sequence Classification

There are usually more unlabeled data than labeled data. Some unlabeled data shares common features with labeled data and also contains extra features which may provide a more comprehensive description of a class. Therefore, by incorporating unlabeled data, sometimes, a more accurate classifier may be built.

For text classification, there is a large amount of unlabeled data. Nigam *et al.* [46] propose a semi-supervised classification method to label documents. Initially, a Naive Bayes classifier is used to classify unlabeled examples in the first round. Then, an Expectation-Maximization (EM) process is utilized to adjust the parameters of the Naive Bayes classifier and re-classify the unlabeled data in an iteration. The process terminates when the classification result is stable. One document may belong to several categories and have multiple labels.

Besides text classification, Zhong *et al.* [66] propose a HMM based semi-supervised classification for time series data. The method uses labeled data to train the initial parameters of a first order HMM, and then uses unlabeled data to adjust the model in an EM process. Wei *et al.* [59] adopt one nearest neighbor classifier for semi-supervised time series classification. The method is designed to handle the situation where only a small amount of labeled data in the *positive* class is available. In the training step, at the beginning, all the unlabeled data is regarded as negative. Then, a 1NN classifier is applied to classify unlabeled data in iteration until the the stopping criteria is met. Wei *et al.* [59] propose a heuristic stopping criteria. In the iteration of labeling more time series as positive, they observe that the minimum distance in the positive class will first decrease and then experience a plateau, and at last decrease again. The iteration will stop when the minimum distance in the positive class starting to decrease after the plateau.

Weston *et al.* [60] propose a semi-supervised protein classification method by using SVM with a cluster kernel. The kernel function between two sequences is defined as the distance between two clusters of sequences. The two clusters are the neighborhoods of the two sequences, and the distance of the two clusters is the average pair-wise inter-cluster distance. The neighborhood of a sequence may contain labeled and unlabeled sequences. By using the cluster kernel, the information of the unlabeled data can be utilized. The results show that by adding unlabeled data, the cluster kernel works better than only using labeled data.

3.3 Sequence Classification with A Sequence of Labels

As discussed in Section 1, for streaming sequence classification, instead of predicting one class label, it is more desirable to predict a sequence of labels. Kadous [24; 23] identifies this problem as *strong sequence classification* task but does not provide a solution for this problem.

A closely related problem considered in natural language processing is called labeling sequences [31; 7; 20]. The task is to label each element in a sequence. For example, given a sentence, where each word is treated as an element, sequence labeling is to assign each word to a category, such as name identity, noun phrase, verb phrase etc. The straightforward solution is to label each element independently. An advanced solution is to consider the labels of the elements in a sequence related to each other. Sequence labeling problem has been solved by using conditional random fields [31]. The problem has also been tackled by other methods, such as using a combined model of HMM and SVM [7] and using a recurrent neural network [20].

4. APPLICATIONS OF SEQUENCE CLASSIFICATION

Sequence classification has a broad range of applications. For different application domains, the classification task has different characteristics. In this section, we summarize and compare major methods applied in several application domains.

4.1 Genomic Data

In recent years, a large amount of DNA and protein sequences are available in public databases, such as GenBank [3], EMBL Nucleotide Sequence Database [1] and the Entrez protein database [2]. To understand the functions of different genes and proteins, sequence classification has attracted a lot of attention in genomic research.

Feature based methods are widely used for genomic sequence classification [35; 12; 13; 52]. k-grams [35; 36; 12] and pattern based feature selection [52] have been used on genomic

sequences. After obtaining features, conventional classifiers, such as SVM [35; 36; 52], rule based classifier [5] and neural networks [10] can be applied to classify genomic sequences.

To measure the distance between two genomic sequences, global alignment and local alignment are widely used methods [32; 45]. After obtaining the distance function, KNN classifier can be used for genomic sequence classification [13]. By using a local alignment kernel [49], SVM can also be used to classify protein sequences without feature selection.

Model based methods, such as profile HMM [56], are also important methods for genomic sequence classification.

Deshpande *et al.* [13] compare the performance of SVM, HMM, and KNN methods for classifying genomic sequence data. They find that SVM outperforms in most cases and feature selection plays an important role in determining accuracies of SVM classifiers. She *et al.* [52] also conclude that SVM is the most effective method for protein classification.

Besides accuracy, other challenges in genomic sequence classification are to speed up classification in order to handling a large amount of data [55] and to train an interpretable classifier to gain knowledge about characteristics of genomic sequences [54].

4.2 Time Series Data

Time series data is an important type of sequence data. In Time Series Data Library [4], time series data across 22 domains, such as agriculture, chemistry, health, finance,industry, are collected. UCR time series data archive [27] provides a set of time series datasets as a benchmark for evaluating time series classification methods.

For simple time series data, to apply feature based methods, the feature selection is a challenging task since we cannot do feature enumeration on numeric data. Therefore, distance based methods are widely adopted to classify time series [61; 26; 59; 48]. It is shown that comparing to a wide range of classifiers, such as neural networks, SVM and HMM, 1-nearest neighbor classifier with dynamic time warping distance is usually superior in classification accuracy [61].

To apply feature based methods on simple time series, usually, before feature selection, time series data needs to be transformed into symbolic sequences through discretization or symbolic transformation [40]. Without discretization, Ye *et al.* [65] propose a method to find time series shapelets and use a decision tree to classify time series. Comparing to distance based methods, feature based methods may speed up the classification process and be able to generate some interpretable results.

Model based methods are also applied to classify simple time series, such as HMM which is widely used in speech recognition [47].

Multivariate time series classification has been used for gesture recognition [24] and motion recognition [38]. The multivariate data is generated by a set of sensors which measure the movements of objects in different locations and directions. For multivariate time series classification, Kadous *et al.* [24] propose a feature based classifier. A set of user-defined meta-features are constructed and a multivariate time series is transformed into a feature vector. Some universal meta-features include the features to describe the trends of increases and decreases and local max or min values. By using those features, multivariate time series with additional non-temporal attributes can be classified by a decision tree. One multivairate time series can be viewed as a matrix. Li *et al.* [31] propose a method to transform a multivariate time series into a vector through singular value decomposition and other transformations. SVM is then used to classify the vectors.

4.3 Text Data

Sequence classification is also widely used in information retrieval to categorize text and documents. The widely used methods for document classification include Naive Bayes [29] and SVM [43]. Text classification has various extensions such as multi-label text classification [67], hierarchical text classification [57] and semi-supervised text classification [46]. Sebastiani *et al.* [51] provide a more detailed survey on text classification .

5. CONCLUSION

In this paper, we provide a brief survey on sequence classification. We categorize sequence data into five subtypes. We group sequence classification methods in feature based methods, sequence distance based methods and model based methods. We also present several extensions of the conventional sequence classification. At last, we compare sequence classification methods applied in different application domains.

We notice that most of the works focus on the classification task on simple symbolic sequences and simple time series data. Although there are a few works on multiple variate time series and complex symbolic sequences, the problem of classifying complex sequence data is still open at large. Furthermore, most of the methods are devoted to the conventional sequence classification task. Streaming sequence classification, early classification, semi-supervised classification on sequence data and the combinations of those problems on complex sequence data which have practical applications, present challenges for future studies.

6. REFERENCES

[1] Embl nucleotide sequence database homepage: `http://www.ebi.ac.uk/embl/`.

[2] Entrez protein database homepage: `http://www.ncbi.nlm.nih.gov/sites/entrez?db=protein`.

[3] Genbank homepage: `http://www.ncbi.nlm.nih.gov/Genbank/`.

[4] Time series data library webpage: `http://www-personal.buseco.monash.edu.au/~hyndman/TSDL/`.

[5] C. C. Aggarwal. On effective classification of strings with wavelets. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 163–172, 2002.

[6] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipmanl. Basic local alignment search tool. *J.Mol.Biol.*, 215:403–410, 1990.

[7] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden markov support vector machines. In *ICML '03: The Twentieth International Conference on Machine Learning*, pages 3–10, 2003.

[8] B. Anibal, S. M. Aranzazu, and R. J. Jose. Early fault classification in dynamic systems using case-based reasoning. *Lecture notes in computer science*, 4177:211–220, 2005.

[9] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian. Traffic classification on the fly. *Computer Communication Review*, 36(2):23–26, 2006.

[10] K. Blekas, D. I. Fotiadis, and A. Likas. Motif-based protein sequence classification using neural networks. *Journal of Computational Biology*, 12(1):64–82, 2005.

[11] B. Cheng, J. Carbonell, and J.Klein-Seetharaman. Protein classification based on text document classification techniques. *Proteins*, 1(58):855–970, 2005.

[12] N. A. Chuzhanova, A. J. Jones, and S. Margetts. Feature selection for genetic sequence classification. *Bioinformatics*, 14(2):139–143, 1998.

[13] M. Deshpande and G. Karypis. Evaluation of techniques for classifying biological sequences. In *PAKDD '02: Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 417–431, 2002.

[14] J. J. R. Diez, C. A. González, and H. Boström. Boosting interval based literals. *Intell. Data Anal.*, 5(3):245–262, 2001.

[15] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. J. Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *PVLDB*, 1(2):1542–1552, 2008.

[16] G. Dong and P. Jian. *Sequence Data Mining*, pages 47–65. Springer US, 2007.

[17] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Chapter 3. Markov Chain and Hidden Markov Model. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, pages 47–65. Cambridge University Press, 1998.

[18] O. Duskin and D. G. Feitelson. Distinguishing humans from robots in web search logs: preliminary results using query rates and intervals. In *WSCD09: Proceedings of the 2009 workshop on Web Search Click Data*, pages 15–19, 2009.

[19] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

[20] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[21] M. P. Griffin and J. R. Moorman. Toward the early diagnosis of neonatal sepsis and sepsis-like illness using novel heart rate analysis. *PEDIATRICS*, 107(1):97–104, 2001.

[22] X. Ji, J. Bailey, and G. Dong. Mining minimal distinguishing subsequence patterns with gap constraints. *Knowl. Inf. Syst.*, 11(3):259–286, 2007.

[23] M. W. Kadous. *Temporal classification: extending the classification paradigm to multivariate time series*. PhD thesis, 2002.

[24] M. W. Kadous and C. Sammut. Classification of multivariate time series and structured data using constructive induction. *Machine Learning*, 58(2-3):179–216, 2005.

[25] L. Kaján, A. Kertész-Farkas, D. Franklin, N. Ivanova, A. Kocsor, and S. Pongor. Application of a simple likelihood ratio approximant to protein sequence classification. *Bioinformatics*, 22(23):2865–2869, 2006.

[26] E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: a survey and empirical demonstration. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 102–111, 2002.

[27] E. Keogh, X. Xi, L. Wei, and C. A. Ratanamahatana. The UCR time series classification and clustering homepage: `http://www.cs.ucr.edu/~eamonn/time_series_data/`, 2006.

[28] E. J. Keogh and M. J. Pazzani. Scaling up dynamic time warping for datamining applications. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 285–289, 2000.

[29] S.-B. Kim, K.-S. Han, H.-C. Rim, and S. H. Myaeng. Some effective techniques for naive bayes text classification. *IEEE Transactions on Knowledge and Data Engineering*, 18(11):1457–1466, Nov. 2006.

[30] D. Kudenko and H. Hirsh. Feature generation for sequence categorization. In *AAAI '98/IAAI '98: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, pages 733–738, 1998.

[31] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, 2001.

[32] T. W. Lam, W.-K. Sung, S.-L. Tam, C.-K. Wong, and S.-M. Yiu. Compressed indexing and local alignment of DNA. *Bioinformatics*, 24(6):791–797, 2008.

[33] T. Lane and C. E. Brodley. Temporal sequence learning and data reduction for anomaly detection. *ACM Trans. Inf. Syst. Secur.*, 2(3):295–331, 1999.

[34] N. Lesh, M. J. Zaki, and M. Ogihara. Mining features for sequence classification. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 342–346, 1999.

[35] C. S. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: A string kernel for SVM protein classification. In *Pacific Symposium on Biocomputing*, pages 566–575, 2002.

[36] C. S. Leslie and R. Kuang. Fast string kernels using inexact matching for protein sequences. *Journal of Machine Learning Research*, 5:1435–1455, 2004.

[37] D. D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *ECML' 98: The 10th European Conference on Machine Learning*, pages 4–15, 1998.

[38] C. Li, L. Khan, and B. Prabhakaran. Real-time classification of variable length multi-attribute motions. *Knowl. Inf. Syst.*, 10(2):163–183, 2006.

[39] M. Li and R. Sleep. A robust approach to sequence classification. In *ICTAI '05: Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence*, pages 197–201, 2005.

[40] J. Lin, E. J. Keogh, L. Wei, and S. Lonardi. Experiencing SAX: a novel symbolic representation of time series. *Data Min. Knowl. Discov.*, 15(2):107–144, 2007.

[41] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1987.

[42] X. Liu, P. Zhang, and D. Zeng. Sequence matching for suspicious activity detection in anti-money laundering. In *PAISI, PACCF and SOCO '08: Proceedings of the IEEE ISI 2008 PAISI, PACCF, and SOCO international workshops on Intelligence and Security Informatics*, pages 50–61, 2008.

[43] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. J. C. H. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.

[44] S. Needleman and C. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J.Mol.Biol.*, 48:443–453, 1970.

[45] L. A. Newberg. Memory-efficient dynamic programming backtrace and pairwise local sequence alignment. *Bioinformatics*, 24(16):1772–1778, 2008.

[46] K. Nigam, A. McCallum, S. Thrun, and T. M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[47] L. Rabiner. A tutorial on HMM and selected applications in speech recognition. In *IEEE*, pages 257–286, 1998.

[48] C. A. Ratanamahatana and E. J. Keogh. Making time-series classification more accurate using learned constraints. In *SDM '04: SIAM International Conference on Data Mining*, 2004.

[49] H. Saigo, J.-P. Vert, N. Ueda, and T. Akutsu. Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11):1682–1689, 2004.

[50] B. Schölkopf, K. Tsuda, and J.-P. Vert. *Kernel Methods in Computational Biology*, pages 171–192. The MIT press, 2004.

[51] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, 2002.

[52] R. She, F. Chen, K. Wang, M. Ester, J. L. Gardy, and F. S. L. Brinkman. Frequent-subsequence-based prediction of outer membrane proteins. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 436–445, 2003.

[53] T. Smith and M. Waterman. Identification of common molecular subsequences. *J.Mol.Biol.*, 147:195–197, 1981.

[54] S. Sonnenburg, G. Rätsch, and C. Schäfer. Learning interpretable SVMs for biological sequence classification. In *RECOMB '05: The Ninth Annual International Conference on Research in Computational Molecular Biology*, pages 389–407, 2005.

[55] S. Sonnenburg, G. Rätsch, and B. Schölkopf. Large scale genomic sequence svm classifiers. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 848–855, 2005.

[56] P. K. Srivastava, D. K. Desai, S. Nandi, and A. M. Lynn. HMM-ModE-Improved classification using profile hidden Markov models by optimising the discrimination threshold and modifying emission probabilities with negative training sequences. *BMC Bioinformatics*, 8(104), 2007.

[57] A. Sun and E.-P. Lim. Hierarchical text classification and evaluation. In *ICDM*, pages 521–528, 2001.

[58] P.-N. Tan and V. Kumar. Discovery of web robot sessions based on their navigational patterns. *Data Min. Knowl. Discov.*, 6(1):9–35, 2002.

[59] L. Wei and E. Keogh. Semi-supervised time series classification. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 748–753, 2006.

[60] J. Weston, C. S. Leslie, D. Zhou, A. Elisseeff, and W. S. Noble. Semi-supervised protein classification using cluster kernels. In *NIPS*, 2003.

[61] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana. Fast time series classification using numerosity reduction. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 1033–1040, 2006.

[62] Z. Xing, J. Pei, G. Dong, and P. S. Yu. Mining sequence classifiers for early prediction. In *SDM'08: Proceedings of the 2008 SIAM international conference on data mining*, pages 644–655, 2008.

[63] Z. Xing, J. Pei, and P. S. Yu. Early classification on time series: A nearest neighbor approach. In *IJCAI'09: Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 1297–1302, 2009.

[64] O. Yakhnenko, A. Silvescu, and V. Honavar. Discriminatively trained markov model for sequence classification. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 498–505, 2005.

[65] L. Ye and E. Keogh. Time series shapeletes: A new primitive for data mining. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009.

[66] S. Zhong. Semi-supervised sequence classification with Hmms. *IJPRAI*, 19(2):165–182, 2005.

[67] S. Zhu, X. Ji, W. Xu, and Y. Gong. Multi-labelled classification using maximum entropy method. In *SIGIR*, pages 274–281, 2005.