Survey on Time-Series Data Classification

Tarek Amr
@gr33ndata

Abstract

Time-series data are every where. They are important in stock market analysis, economics, sales forecasting, and the study of natural phenomena such as temperature and wind speed [Han and Kamber, 2006]. The growing size of such data, as well as its variable statistical nature, make it a challenging problem for data mining algorithms to predict, classify and index. We will focus here on time-series data classification by shedding the light on the researches done in this area.

1 Introduction

Time-series data is a sequence of values obtained at different time points [Lee et al., 2004]. Usually, those values are evenly distributed across the time domain [Dunham, 2002]. Values can be just real numbers as in the case of univariate time-series, or they can be numerous observations received in each time point as in the case of multivariate time series (MTS) [Yang and Shahabi, 2004]. A typical example is when data is collected from different sensors, and data is usually stored in two-dimensional metric to represent the different observations and the number of variables at each of them [Yang and Shahabi, 2004]. Please notice that the terms *time-series data* and *sequential data* will be used interchangeably in this review. While there is no notion of time in sequential data such as text and gene sequences, the order of the data is important when dealing with it, as noted by [Laxman and Sastry, 2006]. Hence, as we will see later on, that many of the approaches meant to deal with one are also suitable for the other.

Statisticians tried to offer new methods for studying time-series data; however, most of time-series data are non-stationary. Non-stationary time-series data, as pointed out by [Priestley, 1988], are defined by their varying statistical properties with time. More flexible models were made to cover these issues, such as *Autoregressive integrated moving average (ARIMA)* [Box et al., 1970], however it still have its limitation by assuming a linear model for the data [Zhang, 2003]. [Priestley, 1988] criticized all previous statistical models, stating that in almost all of them time-series is assumed to confirm to a *linear* model, and is assumed to be either stationary or can be reduced into a stationary one by some transformations.

Data Mining algorithms, on the other hand, were able to offer better solutions. [Zhang, 2003] highlighted that *Artificial Neural Networks (ANN)* doesn't require us to specify the model for the data, but it rather builds the model utilizing the data used in the training process. He added that this makes it easier and more suitable for different kinds of data models. Nevertheless, classical machine learning techniques are designed to deal with static data and cannot simply be used with time-series data. [Vilar et al., 2009] stated that classical algorithms ignore the autocorrelation structure of time-series data. Similarly, [Keogh and Kasetty, 2003] pinpointed that "the high dimensionality, very high feature correlation, and the (typically) large amounts noise that characterize time series data" are the main issues facing classical machine learning algorithms when dealing with time-series data. Hence, many researches try to find new algorithms or adapt the existing ones to suit time-series data. And we are going to focus on the classification algorithms here.

Classification is defined by [Dunham, 2002] as mapping data into predefined classes. The classifier is built using training date and the classes are defined beforehand, hence it is referred to as a supervised learning method, as opposed to clustering. Among the techniques used in classification, [Aggarwal, 2002] listed *decision trees*, *nearest neighbour* and *neural networks*. More on

these techniques will come later on. When it comes to sequential data, [Dong, 2009] explained that classification can either be used to match a whole sequence to a class of other sequences, or to find out if a subsequent belongs to a certain sequence.

1.1 Related Work and Review Structure

As in classification, [Liao, 2005] concluded that all the algorithms designed for clustering time-series data either try to modify the existing static data algorithms to handle the sequential data, or modify the data itself for the existing algorithms to be able to handle it. He added that the ones dealing with time-series data as it is, try in response to find new similarity measures suitable for the sequential nature of the data. Whereas those doing conversion on the sequential data either extract a feature-vector from it to be fed to the classifier (clustering algorithm is his case), or come out with a model for the data. [Keogh and Kasetty, 2003] limited their review to classification algorithms that rely on providing new similarity measures, while [Xing et al., 2010], on the other hand, categorized the classification algorithms into a similar categorization to those of [Liao, 2005]. Similarly, we are going to study the classification algorithms in the following order in section 2:

- Distance-based classification
- Feature-based classification
- Model-based classification

The rest of the report is organized as follows. In section 3 we are going to have a detailed critique of the time-series shapelet-based classifier introduced by [Ye and Keogh, 2009], followed by our conclusion at the end.

2 Time-series classification

2.1 Distance based classification

Classification algorithms such as *k nearest neighbour (kNN)* depend on the the distances between data. And for conventional classification algorithms to work with sequential data, new measurements has to be found to determine the distance between two sequences. [Xing et al., 2010] argues that the choice of distance (similarity) measures play a significant role in the quality of the classification algorithm.

He added that, although *Euclidean distance (ED)* is a widely adopted measurement, it requires the two series in comparison to be of equal length. In additions to this limitation, [Keogh and Kasetty, 2003, Ratanamahatana and Keogh, 2004a] emphasised on its sensitivity to distortion in time. [Sakoe and Chiba, 1978] highlighted that distortion in time-axis is common in speech recognition application where speech rate is not constant. They added that the distortion is also non-linear, hence linear transformation will not be sufficient. [Nerbonne et al., 1999] tried to overcome the time fluctuation by pro-processing (transcribing) the acquired spoken words, however such approach is not practical in most of the cases. In the same fashion, [Chan et al., 2003, Ji et al., 2005] mentioned that in web logs and biomedical data, comparing sequences with gaps is more useful than those without gaps. Thus, elastic similarity measures such as *Dynamic time warping distance (DTW)* were needed to solve this problem.

[Ratanamahatana and Keogh, 2004a] explained DTW as a non-linear mapping between two sequences where the distance between them is minimized. They further explained the algorithm, where "$n * m$" matrix is constructed, and each element in it represents a pairwise distance between points in the two sequences. A path in the matrix is then searched where the total sum of

distances is minimal, which is then returned as the distance between the two strings. Although many researchers [Aach and Church, 2001, Bar-Joseph et al., 2002, Yi et al., 1998] agreed on the superiority of DTW over Euclidean distance, its computational inefficiently is limiting its adoption [Ratanamahatana and Keogh, 2004b]. DTW is calculated using *dynamic programming*, hence has a quadratic time complexity ($O(n*m)$ or $O(n^2)$) [Ratanamahatana and Keogh, 2004a, Xing et al., 2010]. DTW should meet the following local constraints [Sakoe and Chiba, 1978, Keogh and Ratanamahatana, 2005, Yu et al., 2011]:

1. Boundary constraint
2. Monotonicity constraint
3. Continuity constraint

Knowing that the minimal path does fall around the diagonal of the matrix, some researches tried to exploit this fact, in addition to the constrains, in order to speed up the DTW calculations [Xi et al., 2006].

When it comes to symbolic sequences, such as DNA sequences and text strings, alignment-based sequences are preferred [Xing et al., 2010]. [Durbin et al., 1998] states that in evolutionary biology DNA sequences are subject to insertions, deletions and substations. Substitutions for example represent basic mutation processes. Meanwhile, biologists need to find out if sequences are coming from common ancestors by comparing them. He added that both global and local alignments can be measured, whereas local alignment algorithms (such as Smith-Waterman [Smith and Waterman, 1981] and BLAST [Altschul et al., 1990]) try to measures the similarity of sub-sequences rather than for the whole sequence.

[Durbin et al., 1998] describes Needleman-Wunsch global alignment algorithm as follows: A similar matrix to the one described in DTW is built where each axis represents one of the two sequences. The initial value of all the cells is set to zero. Then we fill the matrix applying the formula shown in equation 1, starting from the bottom-right cell, using what is known as *traceback* procedure.

$$F(i,j) = max \begin{cases} F(i-1,j-1) + s(x_i, y_j), \\ F(i-1,j) - d, \\ F(i,j-1) - d \end{cases} \quad (1)$$

Where S(x,y) is the log likelihood ratio of the pair(a,b) occurring as an aligned pair as opposed to an unaligned pair, i.e. a way to the similarity of two characters in biological sequences. And d is defined as the *gap-open penalty*.

[Durbin et al., 1998] highlighted that algorithms (such Needleman-Wunsch [Needleman et al., 1970] and Smith-Waterman [Smith and Waterman, 1981]) are calculated using dynamic programming, hence their complexity is $O(n^2)$. Hence, as noted by [Vinga and Almeida, 2003], more optimum algorithms such as BLAST [Altschul et al., 1990] and FASTA [Pearson et al., 1990] were presented later on. The newer algorithms use heuristic approaches, which means that although they are faster in comparing sequences [Tatusova and Madden, 2006], they do not guarantee finding the optimal score [Durbin et al., 1998]. Additionally, BLAST 2.0 [Tatusova and Madden, 2006] is a tool that utilizes BLAST engine for pairwise sequence comparison, yet it is proposed as an alternative when comparing two sequences that are already known to be homologous.

As mentioned earlier, sequential data can be multivariate. [Yang and Shahabi, 2004] noticed that breaking multivariate time series (MTS) into separate series and processing each one on its own result in overlooking the correlation between those variables. They presented a newer distance-measurement algorithm, Eros (Extended Frobenius norm), in order to deal with MTS.

2.2 Feature based classification

Classical classification algorithms, such as ANN and Decision Trees, do their classification based on feature-set, hence feature-based time-series classification techniques work on transforming the sequential data into feature-set before handing it to the classification algorithms [Xing et al., 2010]. The choice of the appropriate features is the hardest part of this process, and as mentioned by [Eads et al., 2005], there is always trade-off between doing this process manually by domain-experts or having it automated but less accurate in many cases. Patterns and wavelet decomposition, as we will see now, are ways for extracting features from sequential data.

[Laxman and Sastry, 2006] define patterns as short structures that reflects local characteristics of a sequence, such as local spikes in time-series data or the patterns where genes normally appear in a genetic sequence separated by chunks of non-coding DNA. [Xing et al., 2010] added that for patterns to be used in a classification algorithm they should guarantee that they are frequent in at least one class, and should be significantly correlated with it. [Laxman and Sastry, 2006] elaborated that a prototype feature sequences are typically defined for each class after the training process, then new data sequences are classified based on how close they are to that prototype. In a sense, both *patterns* and *models*, (we will come to models later on), are ways to represent data in an abstract way, however, models tend to reflect the global characteristics of the data, while patterns reflect its local ones, They added that the prototype features, acquired during the training phase for each class, normally vary in length. This is why elastic measurement methods, such as the dynamic time warping discussed earlier, are sometimes needed to evaluate how close is a given pattern to our prototypes.

[Ye and Keogh, 2009] noticed that algorithms that try to identify tree-leaves based on their shapes are mislead by the deformation in their shapes due to insects eating parts of them. Instead of relying on the whole shape of the leaves (global features), they selected local features (patterns) that particularly discriminates the leaves from different trees. They converted the shape data into a sequential one. The aim is to find sub-sequences, or shapelets as they called them, that are discriminating between classes. To determine which sub-sequences are to be chosen, they ordered all sequences according to their (Euclidean) distance from all possible shapelets. Then they started to search for a mid-point that divides member-sequences of each class. Having a discriminative approach [Leslie et al., 2002], i.e. binary decisions are taken to tell whether a new sequence belongs to a certain class or not, [Ye and Keogh, 2009] had to use a decision trees in their classifier. The more classes we have, the more branches and split points has the tree. (More on the "shapelets" will come later on in section 3)

Similarly, [Ji et al., 2005] introduced a pattern-extraction algorithm called Minimal Distinguishing Subsequence (MDS). However, MDS allow for gaps with in the sub-sequences, which makes it more suitable to classifying biological sequences as mentioned earlier.

Another feature-extraction technique is to transform the time-series data into the frequency domain, where data dimensionality can be reduced. [Yang and Shahabi, 2004] listed DFT (Discrete Fourier Transform), DWT (Discrete Wavelet Transform) and SVD (Singular Value Decomposition) as examples here. However, [Li et al., 2005] notes that DWT is more common in classification since it preserves both time and frequency characteristics, whereas DFT provides the frequency characteristics only. Such transformation also solves a problem discussed earlier, in 2.1, where we need to study both local and broad trends within the sequential data [Aggarwal, 2002]. DWT transforms the data into different frequency components [Daubechies et al., 1992]. The components with higher order coefficients reflect the global trends of the data, while the ones with lower order coefficients reflect the local trends in it [Aggarwal, 2002].

Kernel methods (KM) are also good in feature extraction, additionally, they can deal with symbol-sequences with different lengths [Watkins, 1999]. Although [Joachims, 1998] was dealing with text data as a bag of words rather than sequential data, he highlighted the ability of kernel methods to deal with textual data regardless of its huge number of features; nor-

mally more than 10k. He was using Support Vector Machine in particular, which is one of the kernel methods. KM calculates the inner product of the input vectors in a high dimensional space [Lodhi et al., 2002]. By doing so, linear decision boundaries can be drawn between the classes [Leslie et al., 2002]. Unlike [Joachims, 1998], [Lodhi et al., 2002] used KM to classify text as sequential data. Like alignment-based distance measures, kernel methods are widely used in biological sequences classification [Liao and Noble, 2003, Zavaljevski et al., 2002].

2.3 Model based classification

According to [Liao, 2005], the model-based methods constructs a model for the data within a cluster (class in our case) and classify new data according to the model that best fits it. He divided the models used in classification into *statistical* and *neural network* ones. According to [Rabiner, 1989], the statistical models such as: *Gaussian*, *Poisson*, *Markov* and *Hidden Markov Models*, are constructed so that they models the probability distribution of the data. [Laxman and Sastry, 2006, Dunham, 2002], on the other hand, divided models into *predictive models* that tries to predict unavailable values of the data using the existing one, and *descriptive models* that tries to find patterns and relationships in the data. We will focus on the predictive models since those are the ones used in classification, especially Markov models which is used a lot in sequence classification applications [Laxman and Sastry, 2006].

Hidden Markov Model (HMM) is defined by [Baldi et al., 1994] as "a set of states S, an alphabet of m symbols, a probability transition matrix $T = (t_{ij})$, and a probability emission matrix $E = (e_{ia})$. When the system is in state i, it has a probability t_{ij} of moving to state j and a probability e_{ia} of emitting symbol a". [Laxman and Sastry, 2006] explained the use of HMM in classification as follows: For each class, a HMM is built using training data from that class, then new patterns are compared to the built models to decide which model (class) fits the new data the best. [Birney, 2001] argues that HMM is more successful in biological sequences classifications, compared to Neural Networks, since it can deal with variable-length sequences, while the other technique require fixed-length inputs. [Rabiner, 1989], on the other hand, pinpointed some of HMM general limitations, such as the assumption that the probability of being in a certain states relies only in the previous state, as well as the assumption that the probabilities of the observations are independent. Similarly, [Graves et al., 2006] criticize the assumption of states probability independence, adding that HMM requires prior domain-specific knowledge to choose the input features.

Generally, artificial neural networks (ANN) are very close to statistical models [Ruck et al., 1990]. [Giles et al., 2001] defines recurrent neural networks (RNN) as special type of ANN, where there is a feedback connection in the network to keep track of its internal state when dealing with new inputs. RNN is suitable for sequential data since, according to [Giles et al., 2001], RNN is capable of modelling the temporal nature of the sequence. Also, [Graves et al., 2006] stated that in contrast to HMM, RNN does not require knowledge of the data. He also claimed that RNN is immune to temporal noise. Nevertheless, as seen earlier, they require fixed-length inputs.

3 Critique of shapelet-based classifier

As we have seen in 2.2, [Ye and Keogh, 2009]'s aim is to find sub-sequences (shapelets) that can be used in building decision trees to classify sequences. To determine the discriminative sub-sequences they used the concept of *information gain* applied by [Olshen and Stone, 1984]. [Quinlan, 1986] describes this concept as follows:

Suppose we have a group of objects p and n which belong to classes P or N respectively. And arbitrary object belongs to P with with the ratio of p to the total number of objects (p+n), similarly it belongs to N with the ratio of n to (p+n). Hence, the information (entropy) of

message source generating such data can be expressed in the following equation:

$$I(p,n) = -\frac{p}{p+n}log\frac{p}{p+n} - \frac{n}{p+n}log\frac{n}{p+n} \tag{2}$$

Now if we use a decision tree to divide a group D into two groups D_1 and D_2, the *information gain (Gain(sp))* for using the value *sp* to split the data D into two subsets is defined as the difference between the entropy before the splitting (I) and the information remaining in the entire dataset after splitting the data ($\hat{I}$). The following equations are taken from [Ye and Keogh, 2009]:

$$Gain(sp) = I(D) - \hat{I}(D) \tag{3}$$

[Ye and Keogh, 2009] stated that the remaining information after the split is obtained by calculating the weighted average entropy of each subset. Defining the fractions of objects in D_1 and $D2$ as $f(D_1)$ and $f(D_2)$ respectively, they ended with the following equation:

$$Gain(sp) = I(D) - (f(D_1)I(D_1) + f(D_2)I(D_2)) \tag{4}$$

Now, to construct a decision tree for our sequential data, we need to find a shapelet and a splitting pint (sp), so that when ordering all the sequences according to their distance from the chosen shapelet, the gain after splitting the sequences using sp will be maximum. Distances between new sequences and the chosen shapelet will then be calculated, and if the distance is within the splitting point they will be considered as members of the same class as that of the shapelet, i.e. discriminative classification [Leslie et al., 2002].

A shapelet is typically much shorter than the sequences. And as highlighted by [Mueen et al., 2011], it creates a compact representation of the class. This means a reduction in the computational memory and time needed during the classification process. [Ye and Keogh, 2009] asserted the complexity of the classification to be $O(\bar{m}l)$, where $\bar{m}$ and l are the the average length of the sequences to be classified and that of shapelet respectively. However, we still need to discuss the following tasks in more details to better understand the performance of the training phase.

- Obtaining all candidate shapelets
- Ordering the sequences based on their distance from the chosen shapelet

A series of m points can be divided into $\frac{m(m+1)}{2}$ possible sub-sequences of lengths 1 to m [Mueen et al., 2011]. In their search of a candidate shapelet, [Ye and Keogh, 2009] can divide all the k sequences in their training sets into all possible sub-sequences. They, however, found it more practical to limit their search to sequences within a certain minimum and maximum range. Nevertheless, the size of the candidate shapelets set is in the range of $\bar{m}^2k$, which means that the complexity of the process increases exponentially with the sequences lengths.

As you have notices, distances here are measured between sequences of different lengths, hence Euclidean distance was redefined by [Ye and Keogh, 2009] as the minimum distance between the smaller sequence and all possible sub-sequences of the same length in the other. Moreover, the distance calculation between the possible shapelets and the sequences in the training set was pointed out by [Ye and Keogh, 2009] to be the most most expensive calculation in the algorithm. Two improvements they presented in the paper later on to speed-up the process: "early abandon" and "early entropy pruning". They first keep on calculating the differences between the corresponding points in the shapelet and each segment of a sequence. However, when the total differences exceeds the minimum distance between the shapelet and another segment of that sequence, the calculation is abandoned, since we are only concerned with the minimum distance here. The other improvement was presented when they noticed that the

entropy calculation is less resource-intensive than the distance calculations. They then decided to calculate the best-case information gain (upper bound) for the remaining sequences after each time they calculate the distance between a new sequence and the shapelet, if the gain is not any better than the maximum gain obtained so far, they stop any further calculations. They tested their classification algorithm on what they said to be the largest class-labelled time-series dataset they are aware of [1]. They concluded that the two improvements combined resulted in a three orders of magnitude speed-up. Later on, [Mueen et al., 2011] presented additional improvement, where they sacrificed additional computer memory resources in exchange of time by caching and reusing repetitive calculations.

Finally, [Ye and Keogh, 2009] justified the shapelet-based classifier by comparing it to 1NN using Euclidean Distance, showing better results for the shapelets. However, [Xing et al., 2011] argued that the algorithms focuses only on the local features and may not be suitable for cases where global features are more differentiating between classes. [Mueen et al., 2011] also argued that a single shapelet might not be enough on its own to differentiate between classes, so they altered [Ye and Keogh, 2009] to use a combination of multiple shapelets, or what they called *logical shapelets*.

4 Conclusion

As we have seen, there are different approaches for classifying sequential data. The nature of data dictate different algorithms sometimes, while the memory and the classification speed are deciding factors some other times.

References

[Aach and Church, 2001] Aach, J. and Church, G. (2001). Aligning gene expression time series with time warping algorithms. *Bioinformatics*, 17(6):495–508.

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

[Altschul et al., 1990] Altschul, S., Gish, W., Miller, W., Myers, E., Lipman, D., et al. (1990). Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410.

[Baldi et al., 1994] Baldi, P., Chauvin, Y., Hunkapiller, T., and McClure, M. (1994). Hidden markov models of biological primary sequence information. *Proceedings of the National Academy of Sciences*, 91(3):1059–1063.

[Bar-Joseph et al., 2002] Bar-Joseph, Z., Gerber, G., Gifford, D., Jaakkola, T., and Simon, I. (2002). A new approach to analyzing gene expression time series data. In *Proceedings of the sixth annual international conference on Computational biology*, pages 39–48. ACM.

[Birney, 2001] Birney, E. (2001). Hidden markov models in biological sequence analysis. *IBM Journal of Research and Development*, 45(3.4):449–454.

[Box et al., 1970] Box, G., Jenkins, G., and Reinsel, G. (1970). *Time series analysis: forecasting and control*, volume 734. Wiley.

[Chan et al., 2003] Chan, S., Kao, B., Yip, C., and Tang, M. (2003). Mining emerging substrings. In *Database Systems for Advanced Applications, 2003.(DASFAA 2003). Proceedings. Eighth International Conference on*, pages 119–126. IEEE.

[1] Synthetic Lightning EMP Classification, http://public.lanl.gov/eads/datasets/emp/index.html

[Daubechies et al., 1992] Daubechies, I. et al. (1992). *Ten lectures on wavelets*, volume 61. SIAM.

[Dong, 2009] Dong, G. (2009). *Sequence data mining*. Springer-Verlag.

[Dunham, 2002] Dunham, M. H. (2002). *Data Mining: Introductory and Advanced Topics*. Prentice-Hall.

[Durbin et al., 1998] Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press.

[Eads et al., 2005] Eads, D., Glocer, K., Perkins, S., and Theiler, J. (2005). Grammar-guided feature extraction for time series classification. In *Proceedings of the 9th Annual Conference on Neural Information Processing Systems (NIPS05)*.

[Giles et al., 2001] Giles, C., Lawrence, S., and Tsoi, A. (2001). Noisy time series prediction using recurrent neural networks and grammatical inference. *Machine Learning*, 44(1):161–183.

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

[Han and Kamber, 2006] Han, J. and Kamber, M. (2006). *Data mining: concepts and techniques*. Morgan Kaufmann.

[Ji et al., 2005] Ji, X., Bailey, J., and Dong, G. (2005). Mining minimal distinguishing subsequence patterns with gap constraints. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE.

[Joachims, 1998] Joachims, T. (1998). Text categorization with suport vector machines: Learning with many relevant features. In *ECML*, pages 137–142.

[Keogh and Ratanamahatana, 2005] Keogh, E. and Ratanamahatana, C. (2005). Exact indexing of dynamic time warping. *Knowledge and information systems*, 7(3):358–386.

[Keogh and Kasetty, 2003] Keogh, E. J. and Kasetty, S. (2003). On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Min. Knowl. Discov.*, 7(4):349–371.

[Laxman and Sastry, 2006] Laxman, S. and Sastry, P. (2006). A survey of temporal data mining. *Sadhana*, 31(2):173–198.

[Lee et al., 2004] Lee, S., Kwon, D., and Lee, S. (2004). Minimum distance queries for time series data. *Journal of Systems and Software*, 69(1-2):105–113.

[Leslie et al., 2002] Leslie, C., Eskin, E., and Noble, W. (2002). The spectrum kernel: A string kernel for svm protein classification. In *Proceedings of the pacific symposium on biocomputing*, volume 7, pages 566–575. Hawaii, USA.

[Li et al., 2005] Li, T., Ma, S., and Ogihara, M. (2005). Wavelet methods in data mining. *Data Mining and Knowledge Discovery Handbook*, pages 603–626.

[Liao and Noble, 2003] Liao, L. and Noble, W. (2003). Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *Journal of computational biology*, 10(6):857–868.

[Liao, 2005] Liao, T. W. (2005). Clustering of time series data - a survey. *Pattern Recognition*, 38(11):1857–1874.

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

[Mueen et al., 2011] Mueen, A., Keogh, E., and Young, N. (2011). Logical-shapelets: an expressive primitive for time series classification. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1154–1162. ACM.

[Needleman et al., 1970] Needleman, S., Wunsch, C., et al. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.

[Nerbonne et al., 1999] Nerbonne, J., Heeringa, W., and Kleiweg, P. (1999). Comparison and classification of dialects. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, pages 281–282. Association for Computational Linguistics.

[Olshen and Stone, 1984] Olshen, L. and Stone, C. (1984). Classification and regression trees. *Wadsworth International Group*.

[Pearson et al., 1990] Pearson, W. et al. (1990). Rapid and sensitive sequence comparison with fastp and fasta. *Methods in enzymology*, 183:63.

[Priestley, 1988] Priestley, M. (1988). Non-linear and non-stationary time series analysis.

[Quinlan, 1986] Quinlan, J. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

[Rabiner, 1989] Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

[Ratanamahatana and Keogh, 2004a] Ratanamahatana, C. and Keogh, E. (2004a). Everything you know about dynamic time warping is wrong. In *Third Workshop on Mining Temporal and Sequential Data*, pages 22–25.

[Ratanamahatana and Keogh, 2004b] Ratanamahatana, C. and Keogh, E. (2004b). Making time-series classification more accurate using learned constraints. In *Proceedings of SIAM international conference on data mining*, pages 11–22. Lake Buena Vista, Florida.

[Ruck et al., 1990] Ruck, D., Rogers, S., Kabrisky, K., Oxley, M., and Suter, B. (1990). The multilayer perceptron as an approximation to an optimal bayes estimator. *IEEE Transactions on Neural Networks*, 1(4):296–298.

[Sakoe and Chiba, 1978] Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49.

[Smith and Waterman, 1981] Smith, T. and Waterman, M. (1981). [a]identification of common molecular subsequences, [o] j. *Molecular Biology*, 147:195–197.

[Tatusova and Madden, 2006] Tatusova, T. and Madden, T. (2006). Blast 2 sequences, a new tool for comparing protein and nucleotide sequences. *FEMS microbiology letters*, 174(2):247–250.

[Vilar et al., 2009] Vilar, J. M., Vilar, J. A., and Díaz, S. P. (2009). Classifying time series data: A nonparametric approach. *J. Classification*, 26(1):3–28.

[Vinga and Almeida, 2003] Vinga, S. and Almeida, J. (2003). Alignment-free sequence comparisona review. *Bioinformatics*, 19(4):513–523.

[Watkins, 1999] Watkins, C. (1999). Dynamic alignment kernels. *Advances in Neural Information Processing Systems*, pages 39–50.

[Xi et al., 2006] Xi, X., Keogh, E., Shelton, C., Wei, L., and Ratanamahatana, C. (2006). Fast time series classification using numerosity reduction. *Pulse*, 100:0.

[Xing et al., 2010] Xing, Z., Pei, J., and Keogh, E. (2010). A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter*, 12(1):40–48.

[Xing et al., 2011] Xing, Z., Pei, J., Yu, P., and Wang, K. (2011). Extracting interpretable features for early classification on time series. *the Proceedings of SDM*.

[Yang and Shahabi, 2004] Yang, K. and Shahabi, C. (2004). A pca-based similarity measure for multivariate time series. In *ACM International Workshop On Multimedia Databases: Proceedings of the 2 nd ACM international workshop on Multimedia databases*, volume 13, pages 65–74.

[Ye and Keogh, 2009] Ye, L. and Keogh, E. (2009). Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 947–956. ACM.

[Yi et al., 1998] Yi, B., Jagadish, H., and Faloutsos, C. (1998). Efficient retrieval of similar time sequences under time warping. In *Data Engineering, 1998. Proceedings., 14th International Conference on*, pages 201–208. IEEE.

[Yu et al., 2011] Yu, D., Yu, X., Hu, Q., Liu, J., and Wu, A. (2011). Dynamic time warping constraint learning for large margin nearest neighbor classification. *Information Sciences*, 181(13):2787–2796.

[Zavaljevski et al., 2002] Zavaljevski, N., Stevens, F., and Reifman, J. (2002). Support vector machines with selective kernel scaling for protein classification and identification of key amino acid positions. *Bioinformatics*, 18(5):689–696.

[Zhang, 2003] Zhang, G. (2003). Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175.