Effect of Mahalanobis Distance on Time Series Classification Using Shapelets

M. Arathi and A. Govardhan

School of Information Technology, JNT University Hyderabad, Hyderabad, India
arathi.jntu@gmail.com, govardhan_cse@yahoo.co.in

Abstract. The sequence of values that are measured at time intervals equally spaced is time series data. Finding shapelets within a data set as well as classifying that data based on shapelets is one of the most recent approaches to classification of this data. In the classification using shapelets, Euclidean distance measure is adopted to find dissimilarity between two time series sequences. Though the Euclidean distance measure is known for its simplicity in computation, it has some disadvantages: it requires data to be standardized and it also requires that the two data objects being compared be of the same length. It is sensitive to noise as well. To overcome the problem, Mahalanobis distance measure can be used. In the proposed work, classification of time series data is performed using time series shapelets and used Mahalanobis distance measure which is the measure of distribution between a point and distribution. Correlations between data set is considered. It does not depend on scale. The cost complexity pruning is performed on decision tree classifier. The Mahalanobis distance improves the accuracy of algorithm and cost complexity pruning method reduces the time complexity of testing and classification of unseen data. The experimental results show that the Mahalanobis distance measure leads to more accuracy and due to decision tree pruning the algorithm is faster than existing method.

Keywords: Time series classification, Shapelets, Mahalanobis distance measure, Decision trees, Information gain, Cost complexity pruning.

1 Introduction

For a decade, there have been a number of papers on time series classification. Spaced at equal time intervals, it is an ordered sequence of values. The analysis of time series data includes various methods that try to perceive such data. That is, time series analysis includes either understanding the underlying data context or forecasting. Applications of its classification is not limited to: scientific investigations, economic and sales forecasting, study of natural phenomena, engineering experiments, analysis of customer behavior, stock market analysis, medical treatments.

The major interest of research in the mining of time series data covers classification, indexing, summarization, clustering and anomaly detection. In classification, an unlabeled time series should be assigned to a class predefined. In indexing, for a given

© Springer International Publishing Switzerland 2015

S.C. Satapathy et al. (eds.), *Emerging ICT for Bridging the Future – Volume 2,*
Advances in Intelligent Systems and Computing 338, DOI: 10.1007/978-3-319-13731-5_57

sequence and measure similarity/dissimilarity, we need to retrieve sequences that are akin to those of time series. In summarization, given a time series object Q containing n data points where n is a large number, an estimation of Q retaining its primary properties is created. In clustering, the time series objects need to be grouped under some similarity/dissimilarity measure. In detecting anomalies, given a time series Q along with some model of normal behavior, all sections of Q containing abnormal behavior are found.

Time series classification algorithms are broadly categorized as distance-based, feature-based and, model-based. In the algorithms that are based on distance, the classification of the data is based on the distance between the objects. The distance measures used to compare the time series data are : Euclidean distance, Dynamic Time Warping (DTW), Longest Common Subsequence (LCSS), Edit Distance on Real sequence (EDR), Edit Distance with Real Penalty (ERP), Sequence Weighted Alignment model (Swale), search of similarity based on Threshold Queries (TQuEST), Spatial Assembling Distance (SpADe). In feature-based classification, the time series sequence is converted to feature vector on which the usual classification methods are applied. Here, feature selection plays an important role. Some of the most popular feature selection/data reduction techniques are: Discrete Fourier Transform (DFT) [1], Discrete Wavelet Transform (DWT) [2], Discrete Cosine Transformation (DCT) [3], Singular Value Decomposition (SVD) [1], Piecewise Aggregate Approximation (PAA) [4], Adaptive PAA [5], ChebyShev Polynomial [6], Symbolic Aggregate Approximation (SAX) [7], and Indexable Piecewise Linear Approximation [8]. The model-based methods construct a model for the data within a class and classify new data according to the model that best fits it. In the classification step, a new sequence is assigned to the class with the highest likelihood.

A most promising recent approach of classifying time series data is finding shapelets within a data set [9]. A shapelet is a subsequence of time series data which represents a particular class. The algorithms that are based on shapelets are interpretable, accurate and faster than existing classifiers [10, 11].

There are two types of classification algorithms: algorithms that consider the whole (single) time series sequence (global features) for classification and algorithms that consider a portion of a single time series sequence (local features) for classification. The shapelets are local features. In classification using shapelets, a shapelet that represents a particular class is identified and then, the classification is done based on the shapelet information. Because shapelets are small in size compared to the original data, algorithms that use shapelets for classification, result in less time and space complexity. Shapelets have successfully been used in many other applications like early classification [12], gesture recognition [13], and as a filter transformation for TSC [14].

For classification with shapelets, decision trees (binary) are used, where each nonleaf node represents a shapelet and leaf nodes represent class labels. To know how well the shapelet classifies the data, information gain [15] is used. Apart from this, the measures, such as, the Wilcoxon signed-rank test [16], Kruskal-Wallis [17], and Mood's Median [18] can also be used. The information gain/entropy measure is a better choice among other measures, because, early entropy pruning can be done to avoid unnecessary computations performed when finding the shapelet.

To compare two time series data, a metric distance measure should be used. A distance measure is metric, if it satisfies the: 1) Positive definiteness 2) Symmetry 3) Triangle Inequality.

The rest of the paper is organized as follows. In Section 2, related work is reviewed. The definition and comparison of the distance measures is discussed in Section 3. The decision tree pruning is discussed in Section 4. The experimental results are discussed in Section 5. The paper is concluded in Section 6.

2 Related Work

The closest work is that of time series classification using shapelets [9]. Here, the authors classify the time series data using shapelets. It generates all possible subsequences of all possible lengths where a subsequence is subset of consecutive values of the time series sequence. Each subsequence is tested to see how well it can classify the data. For this, it generates an object histogram which contains all the time series objects distances to the given subsequence. To find the distance between two time series sequence or between a time series sequence and subsequence, Euclidean distance measure is used. The time series objects in the histogram are in increasing order of distance. An optimization in computing the distance between the time series and subsequence is performed. That is, instead of computing the final distance value between the subsequences of a given time series data and the given subsequence, the distance calculations can be stopped when the partial computation is more than the least distance. This is early abandon [19]. To find the best shapelet, information gain is used.

Another optimization is performed to reduce the time complexity called entropy pruning. This is done during object histogram computation. Once a time series sequence is added to object histogram, it is checked to see if the remaining calculations of other time series objects with the given subsequence can be pruned. For this, the partially computed object histogram is taken. The remaining objects (for which the distance has not been computed to the given candidate) of one class are added to one end of the histogram and the objects of other class are added to the other end of the histogram and vice versa. Now, the information gain is computed. If it is greater than the best known so far, then the histogram computation is continued, otherwise the remaining calculations with the candidate are pruned.

The classification of time series data with shapelets along with their corresponding split point produces a binary decision. Hence, binary decision trees are used. Because one shapelet is not sufficient to classify the entire time series data, a number of shapelets are used which clearly distinguishes one class from other. The shapelets are used along with the distance threshold (split point), which divides the data into two sets. The non leaf nodes of the decision tree specify shapelet and the distance threshold; and leaf nodes specify the class label. To predict class label, the time series sequence is fed into classifier, which moves it from the root to the leaf node. It gives the predicted label. While moving from root to leaf node, the sequence is compared with every shapelet on the path using Euclidean distance measure.

It is possible to have same the best information gain for different subsequences especially for small datasets. Such ties can be broken in favour of the longest subsequence, the shortest subsequence or the subsequence that clearly distinguishes one class from another.

Usually, the time series data are very large with many innumerable values in a sequence. Hence, it is very expensive and difficult to compare two time series. Several methods have been identified for comparing time series sequences. Some of the distance measures as mentioned in Section I are: Euclidean distance, DTW [20, 21], LCSS [22], EDR [23], ERP [24], Swale [25], TQuEST [26], SpADe [27]. The Euclidean distance is summed up by the Euclidean measure between corresponding points in each time series. The metric is most intuitive for comparing time series data. It is very simple to compute with time complexity as O(n). But, the problem with it is that it is sensitive to noise and needs the two sequences of the equal length. It also needs standardization of the time series data, if scales differ. DTW is an elastic measure. The two sequences need not be of equal length. Time shifting is done between the two series by repeating the elements. It is based on dynamic programming, hence has quadratic time complexity. It is sensitive to noise. A threshold value, Q, is introduced by the LCSS technique. The scoring technique handles the noise. If the distance between two sequences is less than Q in each dimension, then they are supposed to match and are given a match reward of 1. If the distance is not less that Q in some dimension, they do not match, and therefore there is no reward. Hence, it is sturdy. It rewards matches, but does not penalize mismatched parts. It is also based on dynamic programming. EDR only scores gaps and mismatches, but do not reward matches. It is robust in presence of noise and time shifting. It is based on dynamic programming. ERP is similar to L_1-norm, but also supports time shifting locally. It is a metric distance measure. It is sensitive to noise. DTW, LCSS, and EDR can handle time shifting locally. But the problem with them is that they are not metric. ERP is based on dynamic programming. Swale is similar to LCSS, but it also penalizes dissimilar parts. TQuEST specifies a threshold query which comprises a query time series TQ and a threshold *th*. The database time series are decomposed into time intervals of subsequent elements where the values are above *th*. The query sequence TQ is also decomposed in such a way. Each interval is considered as a point in two-dimensional space with x as starting time and y as the end of interval. How similar two sequences are, is computed by using Minkowski distance measure. Now, the threshold query returns all the sequences of the database that have a similar interval sequence. All the above distance measures show poor performance if there is shifting and scaling in amplitude dimensions which can be handled by SpADe distance measure.

Our focus is on seeing the performance of Mahalanobis distance measure in time series classification using shapelets and to study the effect of cost complexity pruning on the proposed method. To the best of our knowledge, the proposed method gives more accurate and faster classification of time series data than the existing method.

3 Proposed Method

The techniques of evaluating the similarity of time series sequences have attracted the attention of the database researchers. The selection of a distance function to find the similarity between two sequences is a challenging issue for researcher. There are two numeric measures to compare data objects: similarity & dissimilarity. The similarity measure tells about the extent of similarity of two objects. The value is more when objects have greater similarity. The value is often in the range [0, 1]. The dissimilarity measure specifies the difference between the two data objects. The difference is less when the objects have greater similarity. Zero is often the minimum dissimilarity. In this paper, the dissimilarity/distance measure is used to compare two data objects. Using the Mahalanobis Distance measure instead of the Euclidean distance measure improves the accuracy of the algorithm.

3.1 Euclidean Distance Measure

The Euclidean measure is geometric based on the Pythagorean formula, given as,

$$dist = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2} \tag{1}$$

where n stands for the number of dimensions and p_k, and q_k are the kth components of objects, p and q, respectively.

The advantage of the Euclidean distance measure is its simplicity in computation. But it has some disadvantages. Firstly, it requires that when variables are of different scales, the data need to be standardized. Since, it is incorrect to compare time series data with different offsets and amplitudes, they must be normalized/standardized so that the mean is 0 and standard deviation is 1. The normalization of time series data can be performed by subtracting mean from each value of time series data and dividing the result by standard deviation. Consider a regression problem which makes use of class information regarding, age, test scores as well as time. If all are on different scale, then they cannot be compared. This issue can be surmounted by a normalized Euclidean measure. But it incorporates only variances and not covariances unlike Mahalanobis measure which covers both of them. The next disadvantage is that it requires that both the sequences under comparison must be of the same length, and thirdly, it is sensitive to noise.

3.2 Mahalanobis Distance Measure

The Mahalanobis measure is the one between a point and distribution as presented by P. C. Mahalanobis in the year 1936 [28]. It is a unitless measure used to identify and gauge the similarity of an unknown sample set to a known sample. It differs from the Euclidean by considering the equivalence of the data set and is invariant of scale.

Given a time series $x^{(k)}$, let the ith data point be $x_i^{(k)}$. First, compute the (sample) covariance matrix C = (c_{ij}) of a family of time series $x^{(1)}$, $x^{(2)}$,…, $x^{(N)}$ of lengths n by $c_{ij} = \frac{1}{N-1}\sum_{k=1}^{N}(x_i^{(k)} - \bar{x}_i)(x_j^{(k)} - \bar{x}_j)$. N is the number of instances and $\bar{x}_i$ is the average of the ith data point of the time series $(\bar{x}_i = \frac{1}{N}\sum_{k=1}^{N} x_i^{(k)})$.

The Mahalanobis measure is case of the generalized ellipsoid measure $D_M(x, y) = (x - y)^T M (x - y)$. M is proportional to inverse of covariance matrix i.e., $M \alpha C^{-1}$. Though the Mahalanobis distance measure is often defined by setting M to the inverse of the covariance matrix ($M = C^{-1}$), it is convenient to normalize it when possible so that the determinant of the matrix M is one: M = $(\det(c))^{\frac{1}{n}} c^{-1}$ where n is the time series sequence length. The Mahalanobis distance measure minimizes the sum of distances between time series $\sum_{x,y} D_M(x, y)$ subject to a regularization constraint on the determinant (det(M) = 1). In this sense, it is optimal.

When the covariance is non-singular (det(C) $\neq$ 0), then the covariance is positive definite, and so is the matrix M: it follows that the square root of the generalized ellipsoid distance measure is a metric. That is, $D_M(x, y) = 0 \Leftrightarrow x = y$, it is symmetric, non-negative, and also satisfies the inequality of triangle.

3.3 Euclidean vs. Mahalanobis Distance Measure

Mahalanobis measure takes the co-variances of data objects into consideration leading to elliptic decision boundaries in the 2D case, whereas the Euclidean distance measure leads to circular boundaries. In statistics, the distance is measured by the scale of the data. Standard deviation is scale. For univariate data, an object, one standard deviation away from the mean, is closer to the mean than the one which is five standard deviations away. By computing *z-score*, the distance from the mean can be specified, for normally distributed data. The z-score of x is computed as $z = (x\text{-}\mu)/\sigma$, where μ is mean of the time series data and σ is the standard deviation. This is dimensionless quantity.

The graph in Fig. 1 shows simulated bivariate normal data overlaid with prediction ellipses. The ellipses are 10%, 20%, and so on till 90%. The prediction ellipses are the outlines of the bivariate normal density function. For ellipses near the origin, the probability density is high, and for farther ellipses, it is low.

In Fig. 1, there are 2 red marks. One is at (4, 0), and the other at (0, 2). To see which mark is closer to the origin, let us consider the two distance measures. The Euclidean values are 4 and 2. Hence, according to the Euclidean distance measure, the point at (0, 2) is more nearer to the origin. For this distribution, variance in X axis is more than that in Y axis. Therefore, the point (4, 0) is fewer standard deviations away from the origin than the point (0, 2). Hence, it is less likely to see an observation near (0, 2) than at (4, 0). Thus, according to Mahalanobis distance, the point at (4, 0) is more nearer to the origin than the point at (0, 2).

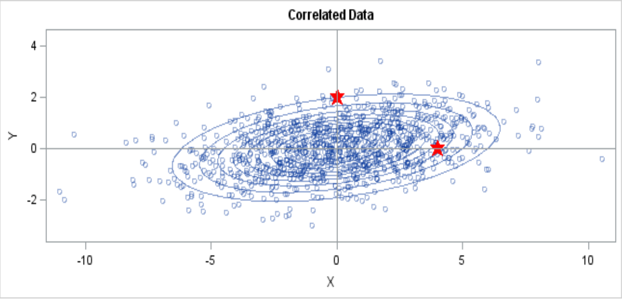

Fig. 1. Bivariate normal data with predicted ellipses

The prediction ellipses is a multivariate generalization of units of standard deviation. The bivariate probability outlines can be used to compare distances to the bivariate mean. Point *a* is nearer to the origin than point *b* if the ellipse containing *a* is enclosed within the ellipse containing *b*.

Mahalanobis measure has the qualities as given below: 1) The variances are different for each dimension. 2) The covariance between variables is considered. 3) For uncorrelated variables with unit variance, its performance is similar to that of Euclidean measure.

4 Decision Tree Pruning

The decision tree classifier is built for time series dataset using shapelets as explained in Section II. It has been observed that while a decision tree is being built, the branches reflect anomalies in the training data owing to noise or outliers. The methods of pruning the tree tackle the data overfitting problem. These methods normally use numerical measures to remove the branches that are least reliable. Pruned tree is small and simple and easy to understand. Pruned trees are fast and good at classifying than unpruned trees.

In decision tree induction process, if a tightly stopping criteria is used, it will lead to small and underfitted trees. But, if loosely halting criteria is used, it will lead to generation of giant decision trees overfitted to the training data set. Many methods to prune decision trees have been introduced to solve the later problem [29]. The two methods of tree pruning are prepruning and post pruning. In the first method, a tree is pruned by halting its construction early. Then the node becomes a leaf holding the most recurrent class among the subset tuples or the probability distribution of those tuples. In post pruning approach, some of the subtrees are withdrawn from a fully generated one. A subtree at a given node is pruned by removing its branches and substituting it with a leaf. That is, given a decision tree classifier C and an inner (non-root, non-leaf) node t. Then pruning of C with respect to t is the deletion of all successor nodes of t in C which makes t a leaf node. The leaf is labeled with the most recurrent class among the subtrees that are substituted. This process is repeated on all nonleaf nodes. The removal of the subtree should not result in reduction of the accuracy of the decision tree. Hence it leads to a smaller and accurate decision tree.

4.1 Cost Complexity Pruning

Cost-complexity pruning is performed in two levels. At the first level, a set of trees DT_0, DT_1, . . . , DT_k are built on the training data. Here, the original tree before pruning is DT_0 while DT_k is the root tree. In the second level, based on its error appraisal, one of these trees is chosen as the pruned tree. The tree DT_{i+1} is procured by substituting one or more of the sub–trees in the preceding tree DT_i with suitable leaves. The sub–trees that are pruned are the ones which get the lowest increase in obvious error rate per pruned leaf:

$$err = \frac{er(prd(DT, dt), Smp) - er(DT, Smp)}{|leaf(DT)| - |leaf(prd(DT, dt))|} \tag{2}$$

where *er*(*DT, Smp*) indicates the error rate of the tree *DT* over the sample *Smp* and |leaf(DT)| indicates the number of leaves in *DT*. *prd(DT,dt)* indicates the tree secured by substituting the node *dt* in *DT* with a suitable leaf.

In the second level, the error of each tree that is pruned DT_0, DT_1, . . . , DT_k is appraised as leading to the selection of the pruned tree that is best ever.

5 Experimental Results

The experiments are conducted on standard datasets such as wheat, mallet, coffee, gun, projectile points, historical documents, beef, car etc. [30]. On all the datasets, our proposed method has shown around 10% – 15% increase in accuracy and 15% – 22% decrease in time complexity.

The wheat dataset contains 775 spectrographs of samples of wheat, which were grown in Canada between 1998 and 2005. There are various kinds of wheat, such as Soft White Spring, Canada Western Red Spring, and Canada Western Red Winter. The wheat grown in a particular year is the class label. For this dataset, the proposed method has shown 12% increase in accuracy as shown in Fig. 2. And there was 20% increase in speed (due to decision tree pruning) during testing phase and classification of unseen data.

There has been extensive study on Gun/NoGun motion capture time series dataset [10], [31]. This data has two classes: Gun and No Gun. The classification algorithm should be able to identify whether the actor is holding gun or not. The difference between the two classes can be identified if the time series data of the actor is observed: how he puts his hand down by his side. The proposed method has shown 8% increase in accuracy for Gun/NoGun problem as shown in Fig. 3. Hence, the proposed method has more accuracy than the existing method. And there was 16% improvement in speed (due to decision tree pruning) during testing phase and during classification of unseen data. Hence, the proposed method is more accurate and fast than existing method.

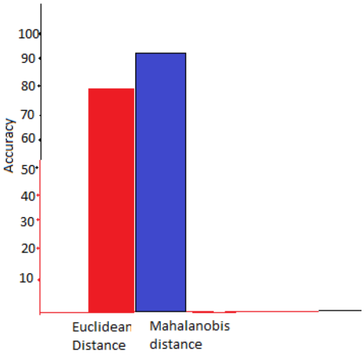

Fig. 2. Accuracy for wheat dataset using Euclidean vs Mahalanobis distance

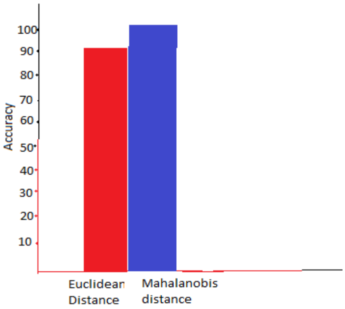

Fig. 3. Accuracy for Gun/NoGun dataset using Euclidean vs Mahalanobis distance

6 Conclusion and Future Scope

The time series dataset is classified using shapelets. The shapelets are time series subsequences and are highly representative of a class. Because one shapelet is not sufficient to classify the data, a number of shapelets are used which clearly distinguishes one class from other. The shapelets are used along with distance threshold, which divides the data into two sets. The decision tree is used as classifier. The non leaf nodes of the decision tree specify shapelet and distance threshold; and leaf nodes specify the class label. To classify a time series data, it is fed into decision tree classifier, which moves it from the root node to leaf node, which in turn gives the predicted class label. While moving from root to leaf node, the time series data is compared with every shapelet on the path using Mahalanobis distance measure. Mahalanobis distance measure is a good choice for classification as it takes the correlation of data items into consideration and is scale in-variant. Hence, it is obvious that Mahalanobis distance measure will give more accurate results. The experimental results have also shown that the distance measure results in more accuracy than the Euclidean distance measure. And we have performed cost complexity pruning on the generated decision tree. The pruning method reduces the size of the decision tree which leads to reduction in time taken in testing phase and also in classification of unseen data. In future, there is scope to compare the proposed method with other distance measures. And also to check how the algorithm will perform on reduced representation of time series dataset. Further, there is also scope to do signature verification using the proposed method.

References

1. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast Subsequence Matching in Time Series Databases. In: SIGMOD Conference (1994)
2. Pong Chan, K., Fu, A.W.-C.: Efficient Time Series Matching by wavelets. In: ICDE (1999)
3. Korn, F., Jagadish, H.V., Faloutsos, C.: Efficiently supporting ad hoc queries in large Datasets of time sequences. In: SIGMOD Conference (1997)
4. Keogh, E.J., Chakrabarti, K., Pazzani, M.J., Mehrotra, S.: Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. Knowl. Inf. Syst. 3(3) (2001)
5. Keogh, E.J., Chakrabarti, K., Pazzani, M.J., Mehrotra, S.: Locally Adaptive dimensionality Reduction for Indexing Large Time Series Databases. In: SIGMOD Conference (2001)
6. Cai, Y., Ng, R.T.: Indexing spatio-temporal trajectories with chebyshev polynomials. In: SIGMOD Conference (2004)
7. Lin, J., Keogh, E.J., Wei, L., Lonardi, S.: Experiencing SAX: a novel symbolic representation of time series. Data Mining Knowledge Discovery 15(2) (2007)
8. Chen, Q., Chen, L., Lian, X., Liw, Y., Yu, J.X.: Indexable PLA for Efficient Similarity Search. In: VLDB (2007)
9. Ye, L., Keogh, E.: Time Series Shapelets: A New Primitive for Data Mining. In: KDD 2009, June 29-July 1 (2009)
10. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures. In: Proc of the 34th VLDB, pp. 1542–1552 (2008)
11. Keogh, E., Kasetty, S.: On the need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. In: Proc. of the 8th ACM SIGKDD, pp. 102–111 (2002)
12. Yu, P., Wang, K., Xing, Z., Pei, J.: Extracting interpretable features for early classification on time Series. In: Proc. 11th SDM (2011)
13. Hartmann, B., Link, N.: Gesture recognition with inertial sensors and optimized DTW prototypes. In: Proc. IEEE SMC (2010)
14. Lines, J., Davis, L., Hills, J., Bagnall, A.: A shapelet transform for time series classification. Tech. report, University of East anglia, UK (2012)
15. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Elsevier Publisher,
16. Wilcoxon, F.: Individual Comparisons by Ranking Methods. Biometrics 1, 80–83 (1945)
17. Kruskal, W.H.: A Nonparametric test for the several sample problem. The Annals of Mathematical Statistics 23(4), 525–540 (1952)
18. Mood, A.M.F.: Introduction to the theory of statistics (1950)
19. Keogh, E., Wei, L., Xi, X., Lee, S., Vlachos, M.: LB_Keogh Supports Exact Indexing of Shapes under Rotation Invariance with Arbitrary Representations and Distance Measures. In: The Proc. of 32nd VLDB, pp. 882–893 (2006)
20. Geurts, P.: Pattern Extraction for Time Series Classification. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, pp. 115–127. Springer, Heidelberg (2001)

21. Keogh, E.J., Ratanamahatana, C.A.: Exact indexing of dynamic time wraping. Knowl. Inf. Syst. 7(3) (2005)
22. Gunopulos, D., Kollios, G.: Discovering similar multidimensional trajectories. In: ICDE (2002)
23. Chen, L., Özsu, M.T., Oria, V.: Robust and fast similarity search for moving object trajectories. In: Sigmod Conference (2005)
24. Chen, L., Ng, R.T.: On the marriage of Lp-norms and edit distance. In: VLDB (2004)
25. Morse, M.D., Patel, J.M.: An efficient and accurate method for evaluating time series similarity. In: SIGMOD Conference (2007)
26. Aßfalg, J., Kriegel, H.-P., Kröger, P., Kunath, P., Pryakhin, A., Renz, M.: Similarity search on time series based on threshold queries. In: Ioannidis, Y., et al. (eds.) EDBT 2006. LNCS, vol. 3896, pp. 276–294. Springer, Heidelberg (2006)
27. Chen, Y., Nascimento, M.A., Oosi, B.C., Tung, A.K.H.: SpADe: On Shape-based Pattern Detection in Streaming Time Series. In: ICDE (2007)
28. Mahalanobis, P.C.: On the generalized distance in statistics. Proceedings of the National Institute of Sciences of India 2(1), 49–55 (1936)
29. Breiman, L., Friedman, J., Olshen, R.A., Stone, C.J.: Classification and regression trees. Wadsworth (1984)
30. Datasets, `http://www.cs.ucr.edu/~eamonn/time_series_data/`
31. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.A.: Fast Time Series Classification using Numerosity Reduction. In: The Proc. of the 23rd ICML, pp. 1033–1040 (2006)
32. Quinlan, J.R.: Simplifying Decision Trees. International Journal of Man-Machine Studies (1987)