High Dimensional Data Mining in Time Series by Reducing Dimensionality and Numerosity

S. U. Kadam\textsuperscript{1} Prof. D. M. Thakore\textsuperscript{2}

\textsuperscript{1}M.E.(Computer Engineering) BVU college of engineering, Pune, Maharashtra, India
\textsuperscript{2}Professors Computer Engineering BVU college of engineering, Pune, Maharashtra, India

Email: \textsuperscript{1}sandiipkadam@gmail.com, \textsuperscript{2}deventhakur@yahoo.com

Abstract

Time series data is sequence of well defined numerical data points in successive order, usually occurring in uniform intervals. In other words a time series is simply a sequence of numbers collected at regular intervals over a period of time. For example the daily prices of a particular stock can be considered as time series data. Data Mining, also popularly known as Knowledge Discovery in Databases (KDD), refers to the nontrivial extraction of implicit, previously unknown and potentially useful information from data in databases. Time series databases introduce new aspects and difficulties to data mining and knowledge discovery due to their high dimensionality nature. By using symbolic representation of time series data we reduce their dimensionality and numerosity so as to overcome the problems of high dimensional databases. We can achieve the goal of time series data mining by introducing a representation of time series that is suitable for streaming algorithms.

1. Introduction

The vast majority of time series data is real valued. Many high level representations of time series have been proposed for data mining. Time series representations can be categorized in Data Adaptive and Non Data Adaptive. While data adaptive can be sub categorized as sorted coefficient, Piecewise Polynomial, Singular value decomposition and non data adaptive can be categorized as Wavelet, PAA, and Spectral. One representation of time series data is to convert the original data into symbolic strings. At first glance this seems a surprising oversight. In addition to allowing the framing of time series problems as streaming problems, there is an enormous wealth of existing algorithms and data structures that allow the efficient manipulations of symbolic representations. Simple examples of “tools” that are not defined for real-valued sequences but are defined for symbolic approaches include hashing, suffix trees, decision trees etc. If the data are transformed into virtually any of the other representations, then it is possible to measure the similarity of two time series in that representation space, such that the distance is guaranteed to lower bound the true distance between the time series in the original space.

Time series representations are suffering from some disadvantages. First, the dimensionality of the symbolic representation is the same as the original data, and virtually all data mining algorithms scale poorly with dimensionality. Second, although distance measures can be defined on the symbolic approaches, these distance measures have little correlation with distance measures defined on the original time series. Third, most of these symbolic approaches require one to have access to all the data, before creating the symbolic representation. This last feature explicitly thwarts efforts to use the representations with streaming algorithms. In addition to allowing the creation of lower bounding distance measures, there is one other highly desirable property of any time series representation, including a symbolic one. Almost all time series datasets are very high dimensional.
This is a challenging fact because all non-trivial data mining and indexing algorithms degrade exponentially with dimensionality. For example, above 16-20 dimensions, index structures degrade to sequential scanning.

Two key aspects for achieving effectiveness and efficiency when managing time series data are representation methods and similarity measures. Time series are essentially high dimensional data and directly dealing with such data in its raw format is very expensive in terms of processing and storage cost. It is thus highly desirable to develop representation techniques that can reduce the dimensionality of time series, while still preserving the fundamental characteristics of a particular data set. In addition, unlike canonical data types, e.g., nominal or ordinal variables, where the distance definition is straightforward, the distance between time series needs to be carefully defined in order to reflect the underlying (dis)similarity of such data. This is particularly desirable for similarity-based retrieval, classification, clustering and other mining procedures of time series.

2. Forward and Backward References

Many techniques have been proposed in the literature for representing time series with reduced dimensionality, such as Discrete Fourier Transformation (DFT), Single Value Decomposition (SVD), Discrete Cosine Transformation (DCT), Discrete Wavelet Transformation (DWT), Piecewise Aggregate Approximation (PAA), Adaptive Piecewise Constant Approximation (APCA), Chebyshev polynomials (CHEB), Symbolic Aggregate approximation (SAX), Indexable Piecewise Linear Approximation (IPLA) and etc. In conjunction with this, there are over a dozen distance measures for similarity of time series data in the literature, e.g., Euclidean distance (ED), Dynamic Time Warping (DTW), distance based on Longest Common Subsequence (LCSS), Edit Distance with Real Penalty (ERP), Edit Distance on Real sequence (EDR), DISSIM, Sequence Weighted Alignment model (Swale), Spatial Assembling Distance (SpADe) and similarity search based on Threshold Queries (TQuEST). Many of these work and some of their extensions have been widely cited in the literature and applied to facilitate query processing and data mining of time series data. However, with the multitude of competitive techniques, we believe that there is a strong need to compare what might have been omitted in the comparisons. Every newly introduced representation method or distance measure has claimed a particular superiority. However, it has been demonstrated that some empirical evaluations have been inadequate and, worse yet, some of the claims are even contradictory. For example, one paper claims “wavelets outperform the DFT”, another claims “DFT filtering performance is superior to DWT” and yet another claims “DFT-based and DWT-based techniques yield comparable results”. Clearly these claims cannot all be true.

A time series $S$ is “a sequence of observed data, usually ordered in time”:

$$S = \{ s_t, t=1,2,..N \}$$

Where $t$ is a time index and $N$ is the number of observations. Time series analysis is fundamental to engineering, scientific, and business endeavors. Researchers study systems as they evolve through time, hoping to discern their underlying principles and develop models useful for predicting or controlling them. Time series analysis may be applied to the prediction of welding droplet releases and stock market price fluctuations.

Traditional time series analysis methods such as the Box-Jenkins or Autoregressive Integrated Moving Average method can be used to model such time series. However, the ARIMA method is limited by the requirement of stationary of the time series and normality and independence of the residuals.

Here we describes a set of methods that reveal patterns in time series data and overcome limitations of traditional time series analysis techniques. The TSDM (Time series data mining) framework focuses on predicting events, which are important occurrences. This allows the TSDM methods to predict irregular time series. The TSDM methods are applicable to time series that appear stochastic, but occasionally (though not necessarily periodically) contain distinct, but possibly hidden, patterns that are characteristic of the desired events.
Many researchers have also considered transforming real valued time series into symbolic representations, noting that such representations would potentially allow researchers to avail of the wealth of data structures and algorithms from the text processing and bioinformatics communities, in addition to allowing formerly “batch-only” problems to be tackled by the streaming community.

3. Time Series Data Mining Functions and Approach

There can be different data mining functions used according to the type of database and techniques used for data mining. We can briefly describe the time series data mining functions which are used for most research work. These functions can be defined as follows.

- Indexing: It can be defined as for a given a query time series Q, and some similarity/dissimilarity measure \( D(Q,C) \), find the most similar time series in database \( DB \).
- Clustering: It is to find natural groupings of the time series in database \( DB \) under some similarity/dissimilarity measure \( D(Q,C) \).
- Classification: It is to assign it to one of two or more predefined classes for given an unlabeled time series Q.
- Summarization: it can be defined as for a given a time series Q containing \( n \) data points where \( n \) is an extremely large number, create a (possibly graphic) approximation of Q which retains its essential features but fits on a single page, computer screen, executive summary, etc.
- Anomaly Detection: Given a time series Q, and some model of “normal” behaviour, find all sections of Q which contain anomalies, or “surprising/interesting/unexpected/novel” behaviour

Based on above functions there can put a generic approach for time series data mining which may take following steps.

1. Create Approximation of the data, which will fit in main memory, yet retain the essential features of interest.
2. Approximately solve the task at hand in main memory.
3. Make(hopefully very few)access to the original data on disk to confirm the solution obtained in step2 modify the solution obtained in step2,or to modify the solution so it agrees with the solution we would have obtained on the original data.

4. Time Series Data Mining by Reducing Dimensionality and Nemerosity

The nature of time series data includes large in data size, high dimensionality and necessary to update continuously. Moreover time series data, which is characterized by its numerical and continuous nature, is always considered as a whole instead of individual numerical field. The increasing use of time series data has initiated a great deal of research and development attempts in the field of data mining. In this technique we use an intermediate representation between the raw time series and the symbolic strings. First transform the data into the Piecewise Aggregate Approximation (PAA) representation and then symbolize the PAA representation into a discrete string. There are two important advantages to doing this, one is directionality reduction and another is lower bounding

4.1 Reducing Dimensionality

Dimensionality reduction is the process of reducing number of random variables or attributes under consideration. Dimensionality reduction can be archived by the method of piecewise aggregate approximation. High-dimensional datasets present many mathematical challenges as well as some opportunities, and are bound to give rise to new theoretical developments. One of the problems with high-dimensional datasets is that, in many cases, not all the measured variables are “important” for understanding the underlying phenomena of interest. While certain computationally expensive novel methods can construct predictive models with high accuracy from high-dimensional data, it is still of interest in many applications to reduce the dimension of the original data prior to any modelling of the data. To reduce the time series from \( n \) dimensions to \( w \)
dimensions, the data is divided into \( w \) equal sized “frames.” The mean value of the data falling within a frame is calculated and a vector of these values becomes the data-reduced representation. The representation can be visualized as an attempt to approximate the original time series with a linear combination of box basis functions as shown in Figure 4.1.

The PAA dimensionality reduction is intuitive and simple, yet has been shown to rival more sophisticated dimensionality reduction techniques like Fourier transforms and wavelets.

![Figure 4.1 PAA representation](image)

We first obtain a PAA of the time series. All PAA coefficients that are below the smallest breakpoint are mapped to the symbol “\( a \),” all coefficients greater than or equal to the smallest breakpoint and less than the second smallest breakpoint are mapped to the symbol “\( b \),” etc. as shown in 4.2 illustrates. Note that in this example the 4 symbols, “\( a \),” “\( b \),” “\( c \)” and “\( d \)” are approximately equiprobable as desired. We call the concatenation of symbols that represent a subsequence a word.

### 4.2 Reducing Numerosity

Numerosity reduction technique replaces the original data by alternative smaller forms of data representation. These methods may be parametric or non-parametric. Now we can say that, given a single time series, our approach can significantly reduce its dimensionality. In addition, the approach of symbolic representation can reduce the numerosity of the data for some applications. Most applications assume that we have one very long time series \( T \), and that manageable subsequence of length \( n \) are extracted by use of a sliding window, then stored in a matrix for further manipulation. Symbolic representation allows for admissible numerosity reduction. This reduction can be explained with an example. Suppose that we are beginning the sliding windows extraction and the first word is \( \text{cabcab} \). If we shift the sliding window by one position, and find the next word is also \( \text{cabcab} \), we can omit it from the Smatrix, without any loss of information, so long as we have kept the index to the first occurrence.

### 5. Analysing Symbolic Time Series Representation

![Figure 5.1: The three different modules](image)

The figure shows the logical flow of time series data for analyzing the symbolic representation of time series in data mining. Raw time series with high dimension is given as input to all three modules. Module one takes input and converts it another PAA representation which is a vector of mean values of each window sliding through raw time series. The PAA format vector serves as input to the discretization process. Which means to have dataset to Gaussian probability. The process also can be called as normalization of data. Finally the discretized data is converted to symbolic data having
input alphabet as equiprobable set of symbols. Thus n dimension data converted to a w dimension where w<n. Module two takes raw time series and applies standard classification and clustering algorithms. The distance measures used give variable results. Also error rate of clustering changes with dimension of dataset. The same is applied to classification algorithm. Module three simply identifies the anomaly within the dataset. Also if any motifs are there will be discovered. Finally analysis of all the results is done.

6. EXPERIMENTAL RESULTS AND ANALYSIS

The raw time series of different dimension is plotted in figure 6.1. The raw time series may have any type of distribution initially.

We first obtain a PAA of the time series. All PAA coefficients that are below the smallest breakpoint are mapped to the symbol “a,” all coefficients greater than or equal to the smallest breakpoint and less than the second smallest breakpoint are mapped to the symbol “b,” as shown in figure 6.3.

Fig 6.1 : Raw time series plotted against dimension

Simply stated, to reduce the time series from n dimensions to w dimensions, the data is divided into w equal sized “frames.” The mean value of the data falling within a frame is calculated and a vector of these values becomes the data-reduced representation. The representation can be visualized as an attempt to approximate the original time series with a linear combination of box basis functions as shown figure 6.2

Fig 6.2 : PAA of time series data (intermediate representation)

Fig 6.3 : Final conversion to symbols. 128 dimension now reduced to 8.
If we compare PAA representation of time series data with symbolic representation we will found that there is finer granularity shown by symbolic representation than PAA. This comparison is shown in figure 6.4.

![Figure 6.4 Comparison of symbolic representation and PAA for showing finer granularity](image)

7. CONCLUSION

We have shown that symbolic representation is competitive with, or superior to, other representations on a wide variety of classic data mining problems, and that its discrete nature allows us to tackle emerging tasks such as anomaly detection and motif discovery. The representation also helps us to reduce the time and space complexities when symbolic representation is used by different data mining algorithms. It may be possible to create a lower bounding approximation of Dynamic Time Warping, by slightly modifying the classic string edit distance. Finally, there may be utility in extending our work to multidimensional time series. The representation helps us to overcome the flaws related with dimensionality, distance measure and data security.

8. REFERENCES


