Similarity Search in Time Series Data Using Time Weighted Slopes

Durga Toshniwal and R. C. Joshi
Department of Electronics and Computer Engineering
Indian Institute of Technology
Roorkee, India
E-mail: durgadec@iitr.ernet.in, joshifcc@iitr.ernet.in

Keywords: Data Mining, Knowledge Discovery, Time Series Data, Similarity Search, Trend Analysis, Variation in Slopes

Received: November 30, 2004

Similarity search in time series data is an area of active interest in the data mining community. In this paper we introduce a novel approach for performing similarity search in time series data. This technique is based on the intuition that similar time sequences will have similar variations in their slopes and consequently in their time weighted slopes. The proposed technique is capable of handling variable length queries and also works irrespective of different baselines and scaling factors.

Povzetek: Opisana je nova metoda rudarjenja podatkov časovnih vrst z iskanjem podobnosti.

1 Introduction

A large portion of scientific and business data stored on computers is comprised of time series data. Some typical examples include stock indices, biomedical data, retail data and atmospheric data. During the past few years, there has been an explosion of research in the area of time series data mining. This includes attempts to model time series data, to design languages to query such data, and to develop access structures to efficiently process queries on such data. The problem of similarity search in time series data is important and non-trivial.

To perform similarity search on time series data, indexing methods that are capable of supporting efficient retrieval and matching of time series data are required. Most of the indexing methods available today for multi-dimensional data such as the R-tree [1] and the R*-tree [2] degrade performance at dimensionalities greater than 8-10 [3] and eventually perform almost like sequential scanning algorithms at high dimensionalities. Thus, to utilize multi-dimensional indexing techniques, it is essential to first perform dimension reduction on time series data. This helps to map the high-dimensional data to a lower dimension space. Then some distance measure such as the Euclidean Distance may be used to calculate the distance and hence the similarity between any two time sequences.

Most of the approaches developed so far for performing similarity search in time series data are based on dimension reduction. Dimension reduction can be performed by several ways. Some commonly used methods for performing dimension reduction include Discrete Fourier Transform (DFT) [4, 5, 6, 7], Discrete Wavelet Transform (DWT) [8, 9, 10, 11, 12], Singular Value Decomposition (SVD) [13] and Piecewise Aggregate Approximation (PAA) [14].

The most frequently used method for dimension reduction is based on the DFT. The DFT is quite suited for naturally occurring sinusoidal signals but it is ill-suited for representing signals having discontinuities.

The Haar wavelet transform is the most commonly used wavelet transform for dimension reduction. But the basis function for Haar is not smooth. Thus the Haar wavelet transform approximates any signal by a ladder like structure. Hence the Haar wavelet transform is not likely to approximate a smooth function using only a few coefficients. So the number of coefficients to be added must be high. Finding wavelets having more continuous derivatives is still an active area of research.

The SVD technique is a data dependent dimension reduction technique. It uses the KL transform for performing dimension reduction. The given data is used to compute basis vectors. So whenever the database is updated, the basis vectors need to be recomputed. The recomputation time may become infeasible for practical purposes especially when the database is very large.

The PAA performs dimension reduction by dividing the time sequences into equal length segments. The corresponding feature sequence comprises of mean values of each segment. But the means representing each segment give only a rough approximation of each time sequence.

In this paper, we introduce a new approach for similarity search in time series databases. We assume that a time series comprises of samples of a single measured variable against time. The proposed approach is based on the observation that similar time sequences will have similar variations in their slopes and hence time weighted slopes. By time weighted slopes we mean that the slope is assigned a weight depending upon its location along the time axis. The technique being proposed involves some data pre-processing that enables it to handle variable length queries. It is also capable of handling global scaling and shrinking of the data and works irrespective of vertical shifts that may exist between the given time sequences. Further, it does not rely on any kind of dimension reduction.

The rest of the paper is organized as follows. Section 2 gives related work. Section 3 describes the proposed approach. In Section 4, we give experimental results to demonstrate the proposed approach by using test data and a case study is included in Section 5. Finally, conclusions and directions for future work are covered in Section 6.

2 Related Work

In this section we discuss some key approaches for performing similarity search in time series data.

Agrawal et al. [4] used the Discrete Fourier Transform to perform dimension reduction. The DFT was used to map the time sequences to the frequency domain and the index so built was called the F-index. For most sequences of practical interest, the low frequency coefficients are strong. Thus the first few Fourier coefficients are used to represent the time sequence in frequency domain. These coefficients were indexed using the R*-tree [2] for fast retrieval. The basis for this indexing technique is Parseval's theorem. The Parseval's theorem guarantees that the distance between two sequences in the frequency domain is the same as the distance between them in the time domain. For a range query the F-index returns a set of sequences that are at a Euclidean Distance $\in$ from the query sequence.

The F-index may raise false alarms but does not introduce false dismissals. The actual matches are obtained in a post-processing step wherein the distance between the sequences are calculated in the time domain and those sequences which are within $\in$ distance are retained and the others are dismissed. The F-index typically handles 'whole matching' queries.

Faloutsos et al. generalized the F-index method in [15] and called it the ST-index. In this technique, subsequence queries are handled by mapping data sequences into a small set of multidimensional rectangles in feature space. These rectangles are indexed using spatial access methods like the R*-tree [2].

A sliding window is used to extract features from the data sequence resulting in a trail in the feature space. These trails are divided into sub-trails which can be represented by their Minimum Bounding Rectangles (MBR). Thus, in place of storing all the points in a trail, only a few MBRs are stored. When a query is presented to the database, all the MBRs intersecting the query region are retrieved. This guarantees no false dismissals but also raises some false alarms as sub-trails that do not intersect the query region but their MBRs are also retrieved.

Chan et al. [8] have proposed to use the DWT in place of DFT for performing dimension reduction in time series data. Unlike the DFT which misses the time localization of sequences, the DWT allows time as well as frequency localization concurrently. The DWT thus bears more information of signals in contrast to DFT in which only frequencies are considered. The approach in [8] employed the Haar Wavelet Transform for mapping high-dimensional time series data to lower dimensions.

A data dependent indexing scheme was proposed in [13] and is known as the SVD method for dimension reduction. The database consists of n-dimensional points. We map them on a k-dimensional subspace, where $k < n$, maximizing the variations in the chosen dimensions. An important drawback of this approach is the deterioration of performance upon incremental update of the index. Therefore the new projection matrix should be calculated and the index tree has to be reorganized periodically to keep up the search performance.

In PAA [14], each time sequence say of length k is segmented into m equal length segments such that m is a multiple of k. If that is not the case, then the sequence is padded with zeros in order to perform the segmentation. The averages of segments together form the new feature vector for the sequence. The correct selection of m is very important because if m is very large, the approximation becomes very rough but if m is very small, the performance deteriorates.

Mostly similarity search methods utilize the Euclidean distance model for calculating the similarity between the query and candidate sequence. According to this model, if the Euclidean Distance $D\ (X,\ Y)$ between two time sequences X and Y of length n is less than a threshold $\in$, then the two sequences are said to be similar. The Euclidean Distance is given as:

$$D\ (X,\ Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (1)$$

A major shortcoming in the Euclidean distance model is that it is not able to handle vertical shifts existing between the time sequences under comparison.

Toshniwal et al. [16] have used the cumulative variation of slopes for computing the similarity in the given time series data. In this technique the data is first pre-processed. Next, each of the time sequence is divided into same number of small, equi-width strips as shown in Figure 1. The cumulative variation in slopes is then computed as the parameter $S\ (Q,\ C)$ where Q and C are the two time sequences under comparison. Mathematically:

$$S(Q, C) = \sqrt{\sum (S_{qj} - S_{cj})^2} \qquad (2)$$

S_{cj} and S_{qj} are the slopes for the j^{th} strip in the candidate time sequence C and the query time sequence Q respectively.

Ideally, for exactly similar time sequences, the parameter S *(Q, C)* would be zero. Practically, the smaller the value of *S (Q, C),* the more is the similarity between the time sequences under comparison. For range queries and nearest-neighbour queries we may choose to have $S(Q, C) \leq /$ where $/$ specifies some degree of tolerance allowed while performing similarity search in the time-series database.

In this paper, we present an approach for similarity search in time series data which is an improvement over [16]. The technique proposed in this paper is also based on the concept that similar sequences will have similar variations in their slopes. In [16], the cumulative difference between the slopes of the query and the candidate time sequences has been computed as given by (2). In the present study, weights have been given to the locations of the slopes along the time axis. In [16] the square of the differences in the slopes has been used for computing the parameter for similarity *S (Q, C)*. However, in this paper, the sign of the slopes have also been accounted for while computing the cumulative variation in slopes by using a cubic function as in (4).

3 Proposed Approach

The cumulative variation in time weighted slopes has been used in this paper for performing similarity search in time series data. Here, we assume that a time series consists of a sequence of real numbers which represent the values of a measured parameter at equal intervals of time. Let the time series database consist of p time sequences designated by $X_1, X_2 \ldots X_p$. Each time sequence X_i in turn can be represented as $< (t_{i1}, y_{i1}), (t_{i2}, y_{i2}) \ldots (t_{in}, y_{in}) >$ where n is the number of samples in the time sequence.

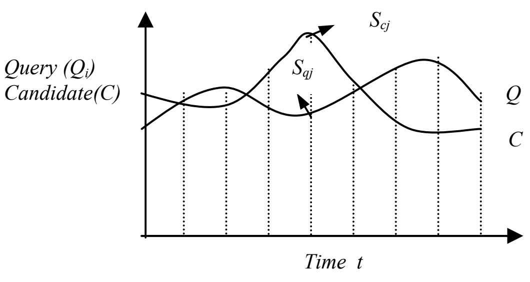


Figure 1: Query and candidate time sequences divided into strips.

In the proposed approach, each of the candidates X_i in the time series database is first scaled along the time axis so that their time axes become equal to some desired time t_d. The selection of t_d is done by the user and may depend on the domain of application of the data. In our technique, scaling along the time axis is done to equalize the time durations of candidates and query. This helps to compare variable length time sequences. For example, a 5-year sales pattern of a Product A can be compared to a 10-year sales pattern of Product B. Another example where scaling can play a crucial role is the comparison of the growth of a tumour for the past 10-months versus the growth of the tumour for past 10-days. In order to avoid any distortions that may arise due to time scaling, the values along the y-axis for each X_i are also scaled proportionately. Thus each transformed X_i denoted by X_i' may be represented as $< (t_{i1}', y_{i1}'), (t_{i2}', y_{i2}') \ldots (t_{in}', y_{in}') >$ where:

$$t_{ij}' = t_{ij} * (t_d / t_{in})$$
$$and\ y_{ij}' = y_{ij} * (t_d / t_{in}) \qquad (3)$$

This is followed by dividing each of the candidate time sequences in the database into same number of small, equi-width strips along the time-axis as shown in Figure 1. Thus each candidate time sequence is divided into say m number of strips.

The same procedure is repeated for any query Q for similarity search. Or in other words, the query is first time scaled to t_d and then scaled proportionately along the y-axis. The resulting sequence is divided into m number of small, equi-width strips. The strips have different heights but same widths along the time-axis as shown in Figure 1.

Finally, we compute the cumulative variation in time weighted slopes between any two sequences Q and C as:

$$WS(Q, C) = \left| \sqrt[3]{\sum_{i=1}^{m} (S_{qj} - S_{cj})^3 * t_j / td} \right| \qquad (4)$$

where S_{cj} and S_{qj} are the slopes for the j^{th} strip (Fig. 1) in the candidate time sequence C and the query time sequence Q respectively :

$$S_{cj} = \{ y_{ic\ (j+1)}'' - y_{icj}'' \} / \Delta t \qquad (5)$$
$$\text{and } S_{qj} = \{ y_{q\ (j+1)}'' - y_{qj}'' \} / \Delta t \qquad (6)$$

We assume in (5) and (6) that the starting and ending coordinates for the j^{th} strip of the candidate are given by $(t'_{icj}, y_{icj}'')$and $(t'_{ic(j+1)}, y_{ic\ (j+1)}'')$. Similarly, the starting and ending coordinates for the j^{th} strip of the query time sequence are given by (t'_{qj}, y_{qj}'') and $(t'_{q(j+1)}, y_{q\ (j+1)}'')$. And Δt is the width of each of the strips and is a constant. The choice of Δt may be user specified or domain specific.

The important thing to note about the selection of Δt is that its value should be optimally selected so that it is neither too small (because that may lead to excessive computations) nor too large (loss of details). The number of equi width strips in the query as well as the candidate time sequences is equal (Fig. 1). As the width of the strips in the query as well as the candidate time sequences are equal, *Δt* is given as:

$$\Delta t = t'_{ic(j+1)} - t'_{icj}$$
$$Or \ \ \Delta t = t'_{q(j+1)} - t'_{qj} \quad (7)$$

The weight associated with the location of the strip along the time axis is given by the factor t_j / t_d . As the number of the strips in the query as well as the candidate sequences is equal, t_j is given as:

$$t_j = t'_{ic(j+1)} = t'_{q(j+1)} \quad (8)$$

Ideally for two exactly similar time sequences, the value of the parameter *WS (Q, C)* must be zero. Practically, the smaller the value of *WS (Q, C),* the more is the similarity between the time sequences under comparison. For range queries and nearest-neighbour queries we may choose to have *WS (Q, C)* $\leq$ / where / specifies some degree of tolerance allowed while performing similarity search in the time-series database.

The cube root of time weighted variations in slopes has been specially chosen to account for the positive or negative sign of the differences between the slopes of the query as well as the candidate time sequences at corresponding time locations while calculating the cumulative variation in slopes. We feel that the inclusion of the sign plays a key role while computing the cumulative variation in slopes between the query as well the candidate time sequences.

The overall strategy thus involves the following steps:
Step 1: Scaling of data along the time-axis to allow variable length queries.
Step 2: Scaling the values of y-ordinates proportionately to avoid any possibility of data distortions arising from step 1.
Step 3: Dividing each time sequences into same number of small, equi-width strips.
Step 4: Computing the parameter *WS (Q, C)* for cumulative variations in time weighted slopes of the two time sequences under comparison. Ideally, it should be zero.

4 Experimental Results

We have evaluated the performance of the proposed technique by considering synthetically generated sample time sequences as the test data. The test data has been designed specially for this purpose so as to include a variety of curves and reverse curves to demonstrate the effectiveness of the proposed approach.

The first set of sample data considered are shown in Figure 2. It comprises of *A1, A2, A3* and *A4*. The dataset has been scaled both along the x-axis and correspondingly along the y-axis taking t_d = 5 and Δt = 0.385 (taken randomly) and the results are shown in Figure 3. Table 1 summarizes the results obtained by computing the parameter *WS (Q, C), S (Q, C)* and *D (Q, C)* taking *A1t* as the query and the others as the candidates. To graphically illustrate the similarity between *A1t, A2t, A3t* and *A4t*, we have shifted *A1t, A2t* and *A4t* vertically so that all of the time sequences have the same initial y-values. This is shown in Figure 4. It is clear from Figure 4 and also from the parameter *WS (Q, C)* computed in Table 1 that *A1t* is most similar to *A2t* and is most dissimilar to *A4t*. Or in other words, *A1* is most similar to *A2* and is very dissimilar to *A4*. In this case, the results of the Euclidean Distance computations and *S (Q, C)* also give the same results as can be seen from Table 1.

Next we have considered a set of reverse curves - *A1R, A2R, A3R* and *A4R* as the sample data as shown in Figure 5. The dataset has been scaled both along the x-axis and correspondingly along the y-axis taking t_d = 5 and the results are shown in Figure 6. Table 2 shows the results obtained by computing the parameter *WS (Q, C)* and *D (Q, C)* taking *A1Rt* as the query and the others as the candidates. To bring out the similarity between *A1Rt, A2Rt, A3Rt* and *A4Rt*, we have shifted *A1Rt, A3Rt* and *A4Rt* vertically so that all of the time sequences have the same initial y-values. This is shown in Figure 7. It is clear from Figure 7 and also from the parameter *WS (Q, C)* computed in Table 2 that *A1Rt* is most similar to *A3Rt* and is most dissimilar to *A4Rt*. Or in other words, *A1R* is most similar to *A3R* and is very dissimilar to *A4R*. As seen from Table 2, the results of the parameter *S (Q, C)* also indicate the same order of similarity for this dataset. But the Euclidean Distance computations do not give correct results.

The dataset considered next comprises of *B1, B2, B3* and *B4* as shown in Figure 8. The dataset has been scaled taking t_d = 5 and the results are shown in Figure 9. Table 3 shows the results obtained by taking *B1t* as the query and the others as the candidates. To show graphically the similarity between *B1t, B2t, B3t* and *B4t*, we have shifted *B2t, B3t* and *B4t* vertically so that all of time sequences have the same initial y-values. This is shown in Figure 10. It can be clearly seen from Figure 10 and also from the parameter *WS (Q, C)* computed in Table 3 that *B1t* is most similar to *B2t* and is most dissimilar to *B4t*. Or in other words, *B1* and *B2* are very similar to each other whereas *B1* is most dissimilar to *B4*. As seen from Table 3, in this case the results of the Euclidean Distance computations as well as the parameter *S (Q, C)* do not provide appropriate similarity comparisions.

The next dataset for similarity search comprises of reverse time sequences *B1R, B2R, B3R* and *B4R* as shown in Figure 11. After scaling, the resulting time sequences are shown in Figure 12 and are denoted by *B1Rt, B2Rt, B3Rt* and *B4Rt.* To graphically show the similarity, we have shifted vertically, *B1Rt, B2Rt* and *B3Rt* so that all of them lie at the same initial y-value as that of *B4Rt* and shown it in Figure 13. The *WS (Q, C), S (Q, C)* and *D (Q, C)* computations are shown in Table 4 taking *B1R* as the query and all others as the candidates. It can be seen clearly from Table 4 and Figure 13 that both the *WS* as well as the *S* parameters indicate that *B1Rt* is most similar to *B3Rt* and is very dissimilar to *B4Rt*. Or in other words, *B1R* is very similar to *B3R* and is dissimilar to *B4R*. The Euclidean Distance is not able to assess the similarity correctly in this case.

The dataset studied next consists of the time sequences *C1, C2, C3* and *C4* and is shown in Figure 14. The pre-processed data *C1t, C2t, C3t* and *C4t* are shown in Figure 15. The results of the computations of *WS (Q, C), S (Q, C)* and *D (Q, C)* are summarized in Table 5. To graphically show the similarity, we have shifted vertically, C*1t, C2t* and C*3t* so that all of them lie at the same initial y-value as that of C*4t* and shown it in Figure 16.

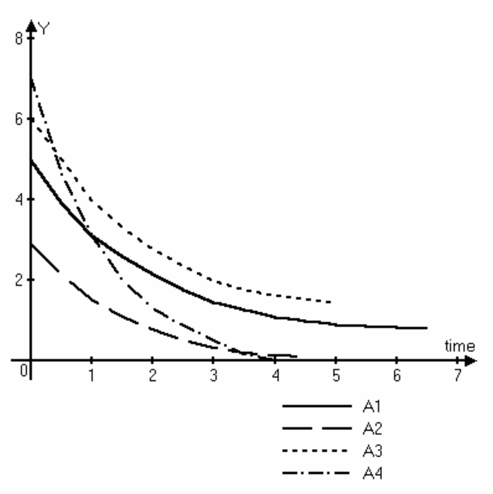


Figure 2: Time series dataset *A*.

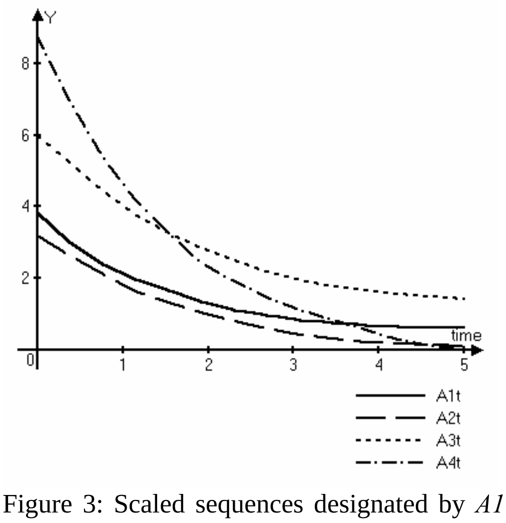


Figure 3: Scaled sequences designated by *A1t, A2t, A3t* and *A4t*.

TABLE 1
PARAMETER *WS (Q, C), S (Q, C)* VERSUS EUCLIDEAN DISTANCE *D (Q, C)*

Sequence Pairs	Parameter *WS*	Euclidean Distance *D*	Parameter *S*
A1t, A2t	0.274	1.60	0.809
A1t, A3t	0.611	5.53	1.241
A1t, A4t	2.058	7.66	4.853

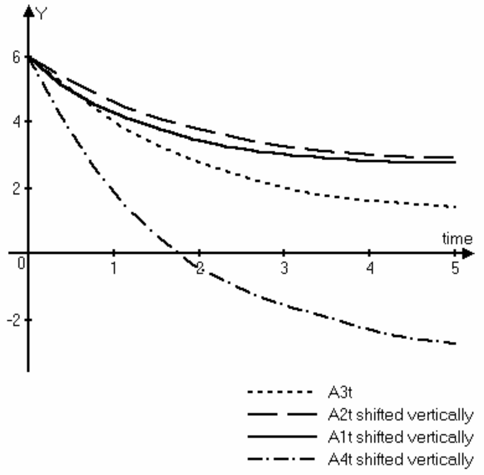


Figure 4: The sequences *A1t, A2t* and *A4t* shifted vertically.

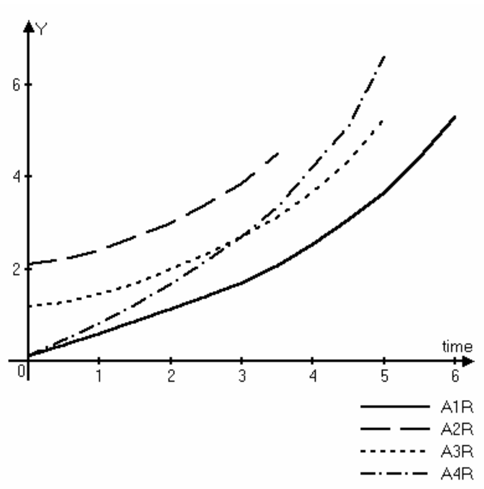


Figure 5: Time series dataset *AR*.

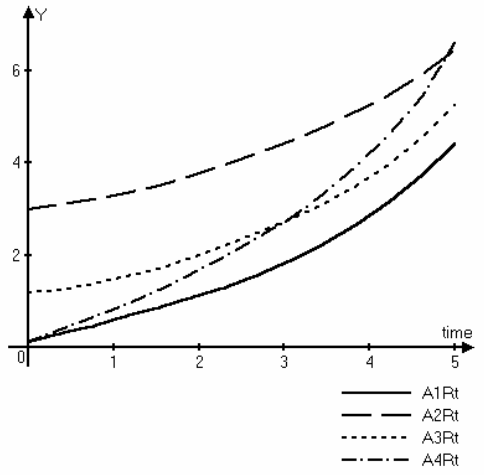


Figure 6: Scaled sequences designated by *A1Rt, A2Rt, A3Rt and A4Rt.*

TABLE 2
PARAMETER *WS (Q, C), S (Q, C)* VERSUS EUCLIDEAN DISTANCE *D (Q, C)*

Sequence Pairs	**Parameter *WS***	**Euclidean Distance *D***	**Parameter *S***
A1Rt, A2Rt	0.566	9.59	0.813
A1Rt, A3Rt	0.151	3.30	0.432
A1Rt, A4Rt	1.359	3.87	1.830

.

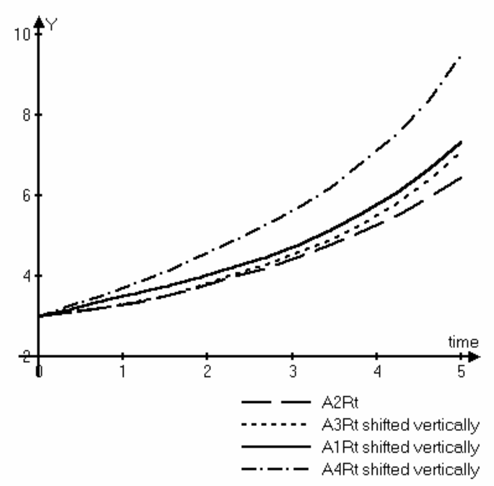


Figure 7: The sequences *A1Rt, A3Rt* and *A4Rt* shifted vertically.

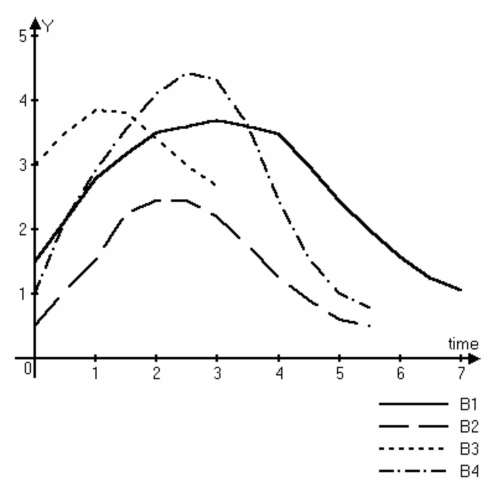
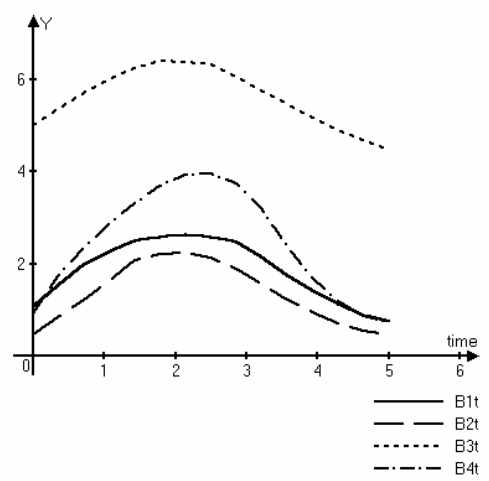


Figure 8: Time series dataset *B*.

It can be concluded from Table 5 and Figure 16 that the query *C1t* is similar to the candidates *C3t, C2t* and *C4t* in that order. While the parameter *S (Q, C)* also indicate the same results, but the Euclidean Distance model gives results which are incorrect. Thus the sequence *C1* is most similar to *C2* and least similar to *C4.*

Finally, we have considered the reverse dataset *CR* as shown in Figure 17. The pre-processed dataset is shown in Figure 18. The results have been computed in Table 6. To graphically show the similarity, we have shifted vertically, C*1Rt, C2Rt* and *C3Rt* so that all of them lie at the same initial y-value as that of C*4Rt* and shown it in Figure 19. It is clear from the parameters *WS (Q, C)* and *S (Q, C)* that *C1R* is most similar to *C3R* and least similar to *C4R.*



. Figure 9: Scaled sequences designated by *B1t, B2t, B3t* and *B4t.*

TABLE 3
PARAMETER *WS (Q, C), S (Q, C)* VERSUS EUCLIDEAN DISTANCE *D (Q, C)*

Sequence Pairs	**Parameter *WS***	**Euclidean Distance *D***	**Parameter *S***
B1t, B2t	0.398	2.06	1.017
B1t, B3t	0.466	14.51	0.886
B1t, B4t	1.243	3.03	2.531

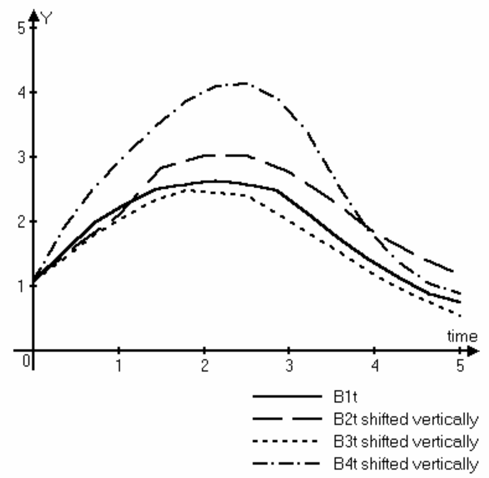


Figure 10: The sequences *B2t, B3t* and *B4t* shifted vertically.

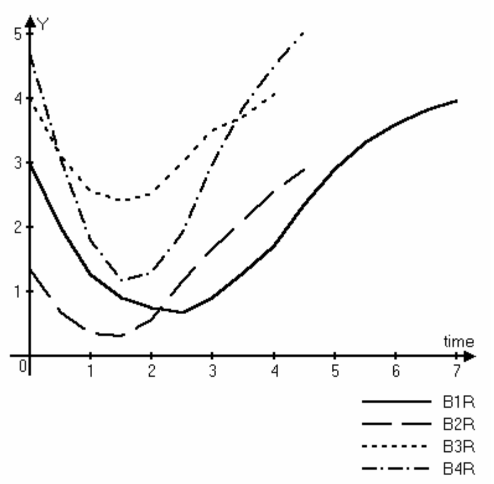


Figure 11: Time series dataset *BR*.

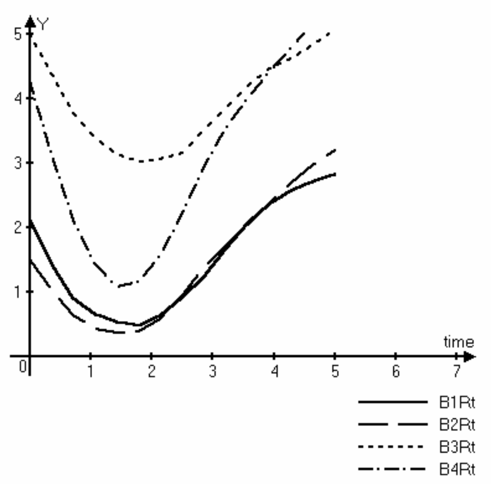


Figure 12: Scaled sequences designated by *B1Rt, B2Rt, B3Rt* and *B4Rt*.

TABLE 4
PARAMETER *WS (Q, C), S (Q, C)* VERSUS EUCLIDEAN DISTANCE *D (Q, C)*

Sequence Pairs	Parameter *WS*	Euclidean Distance *D*	Parameter *S*
B1Rt, B2Rt	0.535	1.00	1.132
B1Rt, B3Rt	0.445	9.43	1.027
B1Rt, B4Rt	1.009	6.65	3.027

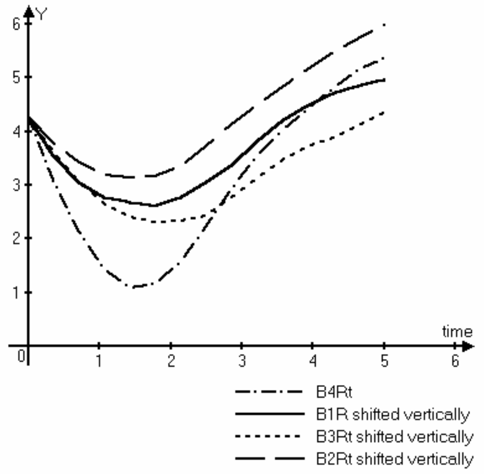


Figure 13: The sequences *B1Rt, B2Rt* and *B3Rt* shifted vertically.

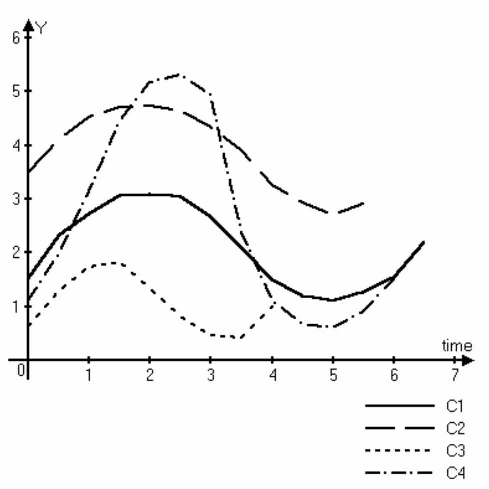


Figure 14: Time series dataset *C*.

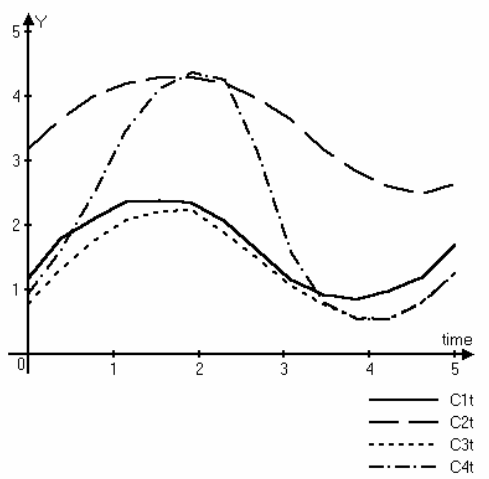


Figure 15: Scaled sequences designated by *C1t, C2t, C3t* and *C4t*.

TABLE 5
PARAMETER *WS (Q, C), S (Q, C)* VERSUS EUCLIDEAN DISTANCE *D (Q, C)*

Sequence Pairs	**Parameter *WS***	**Euclidean Distance *D***	**Parameter *S***
C1t, C2t	1.346	7.22	2.046
C1t, C3t	0.354	1.17	0.876
C1t, C4t	2.571	4.07	4.784

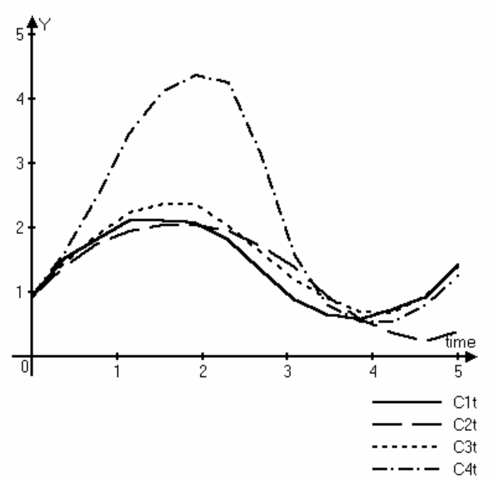


Figure 16: The sequences *C1R, C2R* and C*3R* shifted vertically.

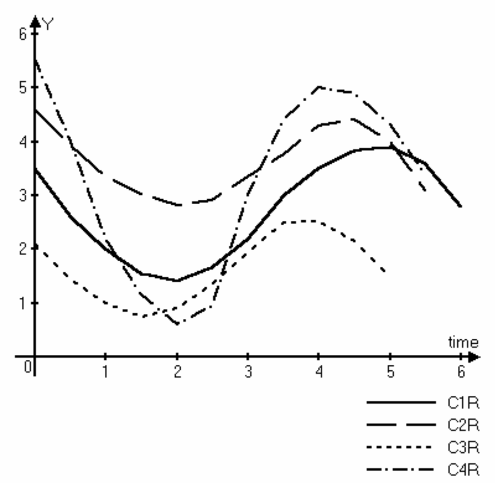


Figure 17: Time series dataset *CR*.

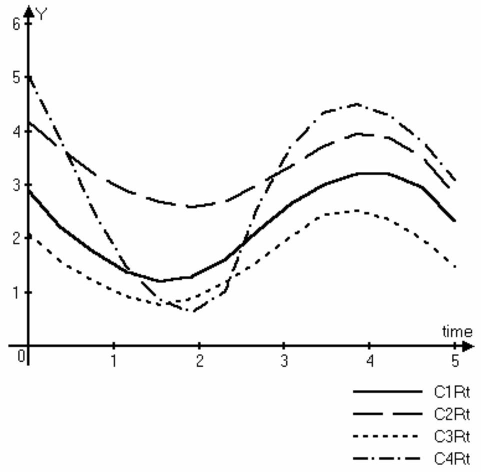


Figure 18: Scaled sequences designated by *C1Rt, C2Rt, C3Rt* and *C4Rt*.

TABLE 6
PARAMETER *WS (Q, C), S (Q, C)* VERSUS EUCLIDEAN DISTANCE *D (Q, C)*

Sequence Pairs	**Parameter *WS***	**Euclidean Distance *D***	**Parameter *S***
C1Rt, C2Rt	0.816	4.04	1.231
C1Rt, C3Rt	0.372	2.46	0.977
C1Rt, C4Rt	1.544	3.99	4.812

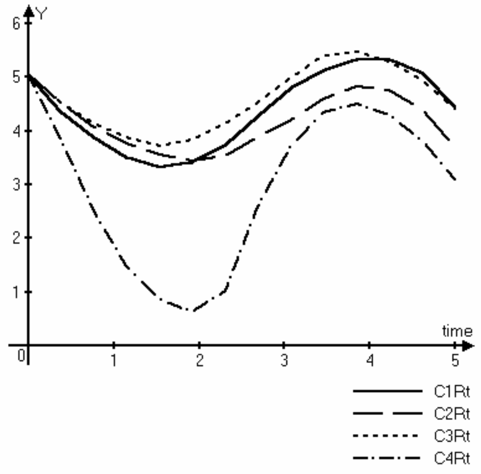


Figure 19: The sequences *C1Rt, C2Rt* and C*3Rt* shifted vertically.

5 Case Study

The case study undertaken in this paper consists of similarity analysis of retail sales data (in millions of dollars) collected on a monthly basis over a period of 11 years (from 01/1992 to 12/2002) for chain retail stores in USA [17]. The length of each time sequence in the retail sales time series database thus consists of 132 datapoints (for each item under sales). We considered sales data of several types of retail businesses as listed in Table 7.

The results of computing the parameter for cumulative variation in time weighted slopes denoted by *WS (Q, C)* given by (4) as compared to the Euclidean Distance given by (1) are shown in Table 8. The Sales at Health and Personal Care Stores has been taken as the query and all others have been taken as the candidate time sequences.

Some of the most similar sequences as evaluated using (4) are shown in Figure 20. It can be concluded from Table 8 that the sales at Health and Personal Care Stores recorded on a monthly basis for a period of 11 years is found to be most similar to the sales at Pharmacies and Drug stores recorded during the same period of time and is found to be the most dissimilar to the sales at New Car Dealers collected during the same period of time.

The results of computing the parameter for cumulative variation in time weighted slopes denoted by *S (Q, C)* given by (2) as compared to the Euclidean Distance given by (1) are shown in Table 9.

TABLE 7
BUSINESSES FOR WHICH RETAIL TIME SERIES DATA HAS BEEN CONSIDERED

S. No.	Description
1	Health and Personal Care Stores (Query)
2	Pharmacies and Drug stores
3	Furniture Stores
4	Jewellery stores
5	Sporting goods, Hobby and Music Stores
6	Household Appliances Stores
7	Men’s Clothing Stores
8	Women’s Clothing Stores
9	Shoe Stores
10	New Car Dealers
11	Used Car Dealers

TABLE 8
PARAMETER *WS (Q, C)* VERSUS *D (Q, C)*

S. No.	Description	Parameter *WS*	Euclidean Distance *D* $(*10^3)$
1	Health and Personal Care Stores (Query)	0	0
2	Pharmacies and Drug stores	362.27	19.86
3	Used Car Dealers	1947.97	79.48
4	Women’s Clothing Stores	2409.69	95.92
5	Shoe Stores	3086.38	105.35
6	Men’s Clothing Stores	3090.50	115.27
7	Furniture Stores	3185.23	52.05
8	Household Appliances Stores	3226.95	114.76
9	Jewellery Stores	3264.25	104.55
10	Sporting goods, Hobby and Music Stores	5341.22	61.52
11	New Car Dealers	7938.09	390.38

The Sales at Health and Personal Care Stores has again been taken as the query and all others have been taken as the candidate time sequences. Some of the most similar sequences evaluated using (2) are shown in Figure 21.

It can be concluded from Table 9 that the sales at Health and Personal Care Stores recorded on a monthly basis for a period of 11 years is found to be most similar to the sales at Pharmacies and Drug stores and the least similar to the sales at New Car Dealers collected during the same period of time. From Table 9 it can be seen that the order of some of the candidate time sequences has changed.

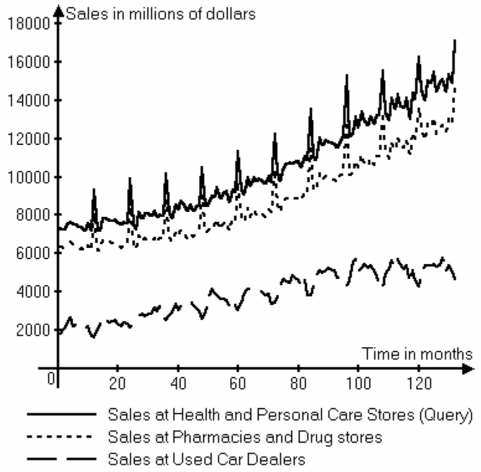


Figure 20: The most similar sequences as indicated by Table 8.

TABLE 9
PARAMETER *S (Q, C)* VERSUS *D (Q, C)*

S. No.	Description	Parameter *S* $(*10^2)$	Euclidean Distance *D* $(*10^3)$
1	Health and Personal Care Stores (Query)	0	0
2	Pharmacies and Drug stores	17.42	19.86
3	Women’s Clothing Stores	57.53	95.92
4	Furniture Stores	71.27	52.05
5	Jewellery Stores	79.03	104.55
6	Shoe Stores	83.32	105.35
7	Men’s Clothing Stores	88.49	115.27
8	Household Appliances Stores	106.22	114.76
9	Used Cars Dealers	126.40	79.48
10	Sporting goods, Hobby and Music Stores	144.22	61.52
11	New Car Dealers	426.39	390.38

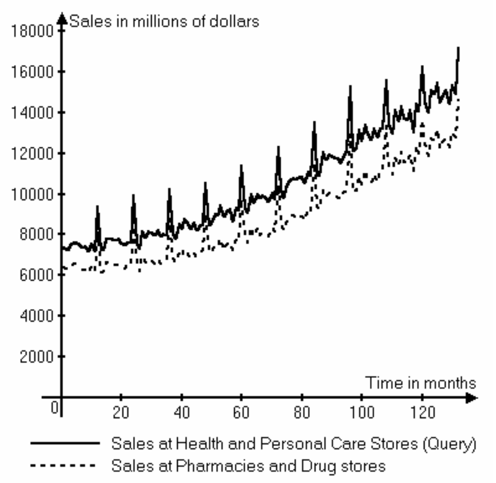


Figure 21: The most similar sequences as indicated by Table 9.

6 Conclusions and Future Work

In this paper, a simple approach for performing similarity search in time series data has been proposed. The given time sequences are pre-processed and brought to the same time range. The y-values are also proportionately scaled to avoid any data distortions that may arise due to scaling along the time axis. The computation of the parameter for cumulative variations in time weighted slopes is done on the pre-processed data. It has been verified by the help of test data that the proposed technique can handle vertical shifts in the time sequence data, global scaling or shrinking of the data as well as variable length queries. No dimension reduction is required in this technique. Euclidean distance model has also been used to compare the test data considered. A case study on retail sales data from stores in USA has been undertaken.

In this approach we have assumed that a time series comprises of samples of a single measured variable against time. In future work, we intend to broaden its scope so that it can handle multivariable time sequences. We also intend to develop alternate parameters using the concept of slopes and time weights for assessing similarity in time series data which may be used individually or in conjunction with each other.

References

[1] A. Guttman (1984) “R-trees: A dynamic index structure for spatial searching,” in *Proceedings of the ACM SIGMOD Conference,* pp. 47-57.

[2] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger (1990) “The R*-tree: An efficient and robust access method for points and rectangles,” in *Proceedings of the ACM SIGMOD Conference,* pp. 322-331.

[3] K.V. Kanth, D. Agrawal, and A. Singh (1998) “Dimensionality reduction for similarity searching in dynamic databases,” in *Proceedings of the ACM SIGMOD Conference.*, pp. 166-176.

[4] R. Agrawal, C. Faloutsos, and A. Swami (1993) “Efficient similarity search in sequence databases,” in *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*, pp. 69-84.

[5] K. Chu and M. Wong (1999) “Fast time-series searching with scaling and shifting,” in *Proceedings of the 18th ACM Symposium on Principles of Database Systems,* pp. 237-248.

[6] C.Faloutsos, H. Jagadish, A. Mendelzon, and T. Milo (1997) “A signature technique for similarity based queries,” in *Proceedings of the International Conference on Compression and Complexity of Sequences,* Positano-Salerno, Italy.

[7] D. Refiei (1999) “On similarity based queries for time series data,” in *Proceedings of the 15th IEEE International Conference on Data Engineering,* pp. 410-417.

[8] K. Chan and A. W. Fu (1999) “Efficient time series matching by wavelets,” in *Proceedings of the 15th IEEE International Conference on Data Engineering,* pp. 126-133.

[9] Y. Wu, D. Agrawal, and A. El Abbadi (2000) “A comparison of DFT and DWT based similarity search in time series databases,” in *Proceedings of 9th ACM International Conference on Information and Knowledge Management,* pp. 488-495.

[10] T. Kahveei and A. Singh (2001) “Variable length queries for time series data,” in *Proceedings of 17th International Conference on Data Engineering*, pp. 273-282.

[11] Z. Struzik and A. Siebes (1999) “The haar wavelet transform in the time series similarity paradigm,” in *Proceedings of Conference on Principles of Data Mining and Knowledge Discovery*, pp. 12-22.

[12] C. Wang and X. S. Wang (2000) “Supporting content based searches on time series via approximation,” in *Proceedings of International Conference on Scientific and Statistical Database Management,* pp. 69-81.

[13] F. Korn, H. Jagadish, and C. Faloutsos (1997) “Efficiently supporting ad hoc queries in large datasets of time sequences,” in *Proceedings of ACM SIGMOD International Conference on Management of Data,* pp. 289-300.

[14] Byoung-Kee Yi and C. Faloutsos (2000) “Fast time sequence indexing for arbitrary Lp norms,” *The VLDB Journal,* pp. 385-394.

[15] C.Faloutsos, M. Ranganathan, and Y. Mano Lopoulos (1999) “Fast subsequence matching in time-series databases,” in *Proceedings of ACM SIGMOD International Conference on Management of Data,* pp. 419-429.

[16] Durga Toshniwal and R. C. Joshi (2004) “ A new approach for similarity search in time series databases based on slopes” , *4th IEEE International Conference on Intelligent Systems, Design and Applications*, Budapest, Hungary, pp. 719- 724.

[17] Economic Time Series Page, http://www.economagic.com