Finding Similarity in Time Series Data by Method of Time Weighted Moments

Durga Toshniwal, Ramesh C. Joshi

Department of Electronics and Computer Engineering
Indian Institute of Technology
Roorkee – 247 667, India

durgadec@iitr.ernet.in, joshifcc@iir.ernet.in

Abstract

Similarity search in time series data is an active area of research in data mining. In this paper we introduce a new approach for performing similarity search over time series data using second moments denoted by us as time weighted moments. This technique is based on the observation that similar time sequences will have their centroids close to each other. The proposed technique is capable of handling variable length queries. It works irrespective of global scaling and shrinking of time series data and can also handle different baselines.

Keywords: Data mining, Knowledge discovery, Temporal data, Time series data mining, Similarity search in time series data, Information retrieval.

1 Introduction

Much of the scientific and business data stored in computers is time series data. Some typical examples include financial data, biomedical data, and climate data. In the last decade, there have been several attempts to model time series data, design query languages for it, and to develop access structures for efficient storage and retrieval of time series data. The problem of similarity search in time series data is non-trivial.

Similarity search on time series data requires indexing methods that are capable of supporting efficient retrieval and matching of time series data. Most of the indexing methods for multi-dimensional data such as the R-tree (Guttman 1984) and the R*-tree (Beckmann 1990) degrade performance at dimensionalities greater than 8-10 (Kanth, Agrawal, and Singh 1998) 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. Dimension reduction maps high-dimensional data to a lower dimension space. Next, 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 for performing similarity search in time series data developed so far rely on dimension reduction. This may lead to loss of information of some kind.

There are several ways of performing dimension reduction on time series data. Some of the commonly used methods include Discrete Fourier Transform (DFT) (Agrawal, Faloutsos, and Swami, 1993, Kamel, and Faloutsos 1994, Chu, and Wong 1999, Faloutsos, Jagadish, Mendelzon and Milo, T. 1997), Discrete Wavelet Transform (DWT) (Refiei 1999, Chan and Fu 1999, Kahveei and Singh 2001, Struzik and Siebes 1999), Singular Value Decomposition (SVD) (Korn, Jagadish and Faloutsos 1997) and Piecewise Aggregate Approximation (PAA) (Yi and Faloutsos 2000).

The DFT is well suited for naturally occurring signals, which are sinusoidal in nature, but is ill suited for others.

The most commonly used Wavelet Transform for dimension reduction is the Haar wavelet transform. The basis function for Haar is not smooth and as a result the Haar wavelet transform approximates any signal by a ladder like structure. Thus 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.

The SVD technique uses the KL transform for performing dimension reduction. The key weakness of this approach is that the SVD is data dependent. This means that it uses the dataset to determine new basis vectors. So it has to be recomputed whenever a database item is updated.

In case of PAA, the time sequence is divided into equal length segments. The corresponding feature sequence comprises mean values of each segment. But the means representing each segment give only a rough approximation of each time sequence.

In this paper, 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. We introduce a new technique for similarity search in time series databases using second moments also denoted as time weighted moments in this paper. The introduced technique is based on the assumption that similar time sequences will have their centroids close to each other. In the approach proposed here, we will use second moments to obtain the centroids. The second moments have been used as they provide weights to the locations of the measured parameter along the x-axis (time axis in this case). This helps to exaggerate the similarity (dissimilarity) measure computed in our approach. In our approach, we use a simple procedure to equalize the lengths of different time sequences along the time axis without distorting the data. Thus the proposed technique is capable of handling variable length queries on time series data. It can also handle time scaling, amplitude scaling or a combination of both. The performance of the proposed technique is independent of the number of datapoints in the candidate or query time sequences.

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 of the proposed technique using test data and Section 5 covers the case study. Finally, conclusions and directions for future work are covered in Section 6.

2 Related Work

In this section we briefly discuss some key approaches for performing similarity search in time series data based on dimension reduction.

Agrawal, Faloutsos and Swami (1993) 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 (Beckmann, Kriegel, Schneider and Seeger 1990) for fast retrieval. The basis for this indexing technique is Parseval's theorem.

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 ∊ distance are retained and the others are dismissed. The F-index typically handles whole matching queries.

The F-index method was generalized by Faloutsos, Ranganathan and Lopoulos (1994) and called 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 (Beckmann, Kriegel, Schneider and Seeger 1990).

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.

Chan and Fu (1999) 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 used by Chan and Fu (1999) employed the Haar Wavelet Transform for mapping high-dimensional time series data to lower dimensions.

A data dependent indexing scheme was proposed by Yi and Faloutsos (2000) 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 (Faloutsos, Ranganathan, and Lopoulos 1994) each time sequence say of length *k* is segmented into *m* equal length segments such that *m* is a multiple of *k*. 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.

3 Proposed Approach

We propose to use centroids for similarity search in time series data. Our approach is based on the idea that similar time sequences will have their centroids close to each other. Ideally, for a exact match between the query and candidate time sequences, the distance between their centroids would be zero. In the following section, we define second moments and centroids.

3.1 Second Moments (Time Weighted Moments) and Centroids

The second moment of an area *A* about the y-axis is given as (Streeter and Wylie 1997):

$$I_y = \int_A x^2 \ dA \tag{1}$$

The integration is carried out over the area. The centroid axis perpendicular to the y-axis is obtained as:

$$t_c = \frac{1}{A} \int_A x^2 \ dA \tag{2}$$

Similarly, the second moment of area *A* about the x-axis is given by:

$$I_x = \int_A y^2 \ dA \tag{3}$$

and the corresponding centroid axis perpendicular to the x-axis is given by:

$$y_c = \frac{1}{A} \int_A y^2 \ dA \tag{4}$$

The point of intersection of the centroidal axes is called the centroid of the area.

In the proposed approach, we perform some data pre-processing steps so as to facilitate centroid computation. It is assumed here that the time series database consists of n time sequences designated by X_1, X_2… X_n. Each time sequence X_i in turn can be represented as < (t_{i1}, y_{i1}), (t_{i2}, y_{i2})… (t_{in}, y_{in}) >.

The first step in data pre-processing involves scaling of each of the candidates X_i in the time series database along the time axis. This is done to equalize their time axes to some desired value say t_d. Thus their time axes become equal. 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 help compare variable length time sequences. For example, a 5-year sales pattern of a Company A can be compared to a 10-year sales pattern of Company 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 aforesaid scaling along the time-axis, the values along the y-axis for each X_i are also scaled proportionately. Each transformed X_i denoted by X_i'' may be represented as <(t_{i1}', y_{i1}'), (t_{i2}', y_{i2}')… (t_{in}', y_{in}') > where:

$$t_{ik}' = t_{ik} * (t_d / t_{in}) \text{ and } y_{ik}' = y_{ik} * (t_d / t_{in}) \qquad (5)$$

The next step involves vertical shifting of all the time sequences (candidates as well as the query time sequences) so that their initial y-coordinate values coincide. This may result in vertically shifting up of some time sequences and downwards of some others. This step is necessary for uniform basis of comparison of moments and centroids for the time sequences under analysis. Thus X_1 becomes X_1' where X_1' is represented as < $(t_{i1}, y_{i1} + y_s)$, $(t_{i2}, y_{i2} + y_s)$… $(t_{in}, y_{in} + y_s)$ > where y_s is the vertical shift which may be positive (moving the sequence up) or negative (moving the sequence down).

To facilitate moment computation we want that $(y_{ij} + y_s)$ be positive always. Or in other words we want the values of the measured parameter along the y-axis to be positive always. So we subtract from the y-coordinate values of all the time sequences under analysis, a quantity denoted by y_{min} where y_{min} is the minimum value for $(y_{ij} + y_s)$ across all the time sequences under analysis. Thus X_1' becomes X_1'' where X_1'' is given by < $(t_{i1}, y_{i1} + y_s - y_{min})$, $(t_{i2}, y_{i2} + y_s - y_{min})$… $(t_{in}, y_{in} + y_s - y_{min})$ > . The result is that we get values for the y-coordinate which are all positive for each of the time sequence being analyzed. Also, each time sequence has the same initial value for the y-coordinate. This completes the pre-processing of the time series data in our approach.

For simplicity of notations, we will henceforth designate the transformed y-coordinate values as y_1 for $(y_{i1} + y_s - y_{min})$, y_2 for $(y_{i2} + y_s - y_{min})$ and so on.

The centroids of the time sequences are now calculated for assessing the similarity of the time sequences under question. For this we need to calculate moments as in (1) and then the centroid as given by (2) can be computed as $(1/A)\sum t^2 \; \Delta A \cdot$

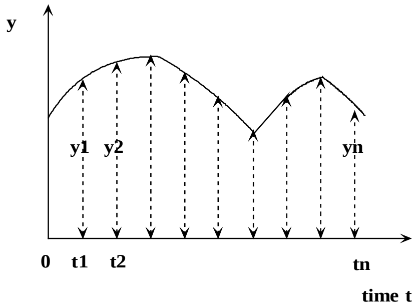

Figure 1: Division of the transformed time series into *n* equi-width strips each having width *D t*

For this we divide each time sequence into small equi-width strips as shown in Figure 1. Let the total moment for each time sequence be denoted by M_x where M_x is given by:

$$M_x = \Delta t \,.\, y_1 .\, t_1^2 + \Delta t .\, y_2 .\, t_2^2 + \ldots + \Delta t \,.y_n .\, t_n^2$$

Or simply, $M_x = \Delta t \sum y_i \,.t_i^2$ *for i = 1 to n* (6)

Here, width of each strip is given by Δt and the area of the strip is given as $y_i . \Delta t$ where Δt is a constant and is given by $\Delta t = t_{i+1} - t_i$. Thus the centroidal axis along the y-axis is given by:

$$t_c = (\Delta t \sum y_i \,.t_i^2) / (\Delta t \sum y_i \, t_i) \text{ for } i = 1 \text{ to } n \qquad (7)$$

Or, $t_c = \sum y_i \,.t_i^2 / \sum y_i \, t_i$

Similarly using (3) and (4), we obtain the centroidal axis parallel to the x-axis denoted by y_c .The intersection of the centroidal axes denoted by t_c and y_c constitutes the centroid for any given time sequence. The closer the centroids are to each other, the similar are the time sequences.

The overall strategy thus involves the following steps:

Data pre-processing: Scaling of data along the time-axis and correspondingly scaling the values of y ordinate to avoid any possibility of data distortions (This is done to allow variable length queries). Vertically shifting the time sequences (candidates as well as the query time sequence) so that their initial y-coordinate values coincide (This brings the time sequences to the same baseline). Making the values of the measured parameter along the y-axis positive (This facilitates centroid computation).

Centroid computation: Computation of moments and centroids using the pre-processed data obtained from previous step for assessing similarity in time series data.

4 Experimental Results

We have evaluated the performance of the proposed technique by considering synthetic sample time sequences as the test data. The data used in this section has been designed especially so that it includes a variety of similar and reverse trend curves. Some of the curves considered here are enlarged or compressed versions of others. Such curves will help demonstrate the ability of our method to handle global scaling or shrinking of time sequences. We have generated 6 sets of synthetic curves namely A, B, C, AR, BR, and CR (the latter three are reverse of the former ones).

The first set of sample data considered are shown in Figure 2. The dataset has been scaled both along the x-axis and correspondingly along the y-axis taking $t_d = 5$ (shown in Figure 3). This value for t_d has been selected randomly. The scaled time sequences are designated by A1T, A2T, A3T, and A4T. All the time sequences have been vertically shifted so that their initial value along the y-axis is say 3.2 which is initial value for A2T. The resulting time sequences are designated by A1T', A2T', A3T' and A4T' and are shown in Figure 4. Finally, the data sequences of Figure 4 have been vertically shifted as shown in Figure 5 so that all of them have their y-values positive. In this case, the minimum value of y across all the time sequences is –5.5. So the sequences have been vertically translated by +5.5 to obtain A1F, A2F, A3F, and A4F. Taking A1F as the query, it is clear visually from Figure 5 that A2F is nearest to A1F and A4F is farthest.

The centroid calculations have been shown in Table 1. The distance of the centroid of A1F from A2F, A3F and A4F quantitatively confirm the conclusions made from Figure 5. Thus we may conclude that A2 is most similar to A1 whereas A4 is most dissimilar to A1.

The next time series dataset under consideration has been shown in Figure 6. The dataset has been scaled both along the x-axis and correspondingly along the y-axis taking $t_d = 5$ (shown in Figure 7). Next, they have been shifted vertically as shown in Figure 8 so that all of them have their initial y-value as 3 which is the initial y-value for A2RT.

Finally all the time sequences in dataset AR have been translated vertically by +0.24 as the minimum y-value across all sequences in this dataset is –0.24. This is shown in Figure 9. Table 3 shows the centroid calculations for this dataset and Table 4 shows the distance computations between the centroids. It is evident from Figure 9 that A1RF (taken as query) is closest to A3RF. Table 4 has confirmed the same conclusions. Thus it can be concluded that by our approach A3R is most similar to A1R (taken as query) as their centroids are very close to each other. On the contrary, A1 is very dissimilar to A1R as indicated by their centroids.

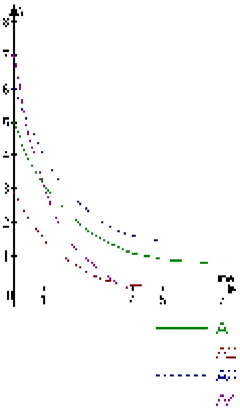

Figure 2: Time series dataset A

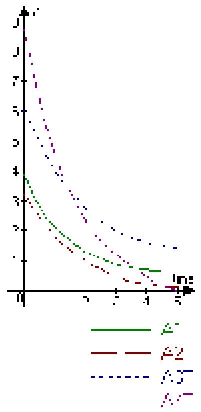

Figure 3: Scaled time series dataset A with t_d = 5

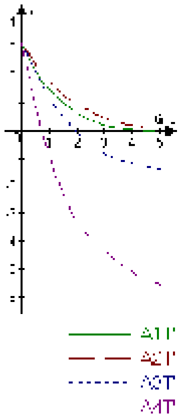

Figure 4: Scaled time series dataset A with t_d = 5 and initial value of y = 3.2

Sequence	t_c	y_c
A1F	3.38	9.62
A2F	3.43	10.06
A3F	3.32	7.57
A4F	2.11	1.95

Table 1: Centroids for the time sequences A

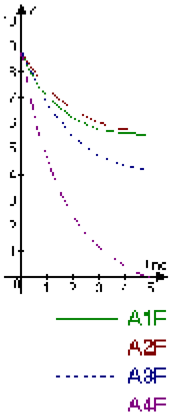

Figure 5: Final pre-processed time series dataset A

The next time sequence dataset B is shown in Figure 10. Its finally pre-processed form is shown in Figure 11. Table 5 shows the centroid computations and Table 6 shows the distance of the centroid of B1F (taken as query) from B2F, B3F and B4F.

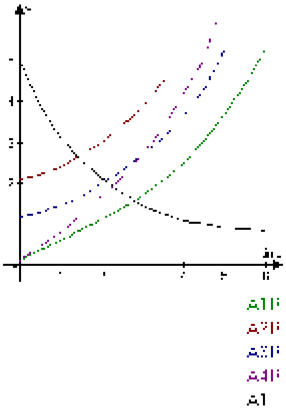

Figure 6: Time series dataset AR (A Reverse)

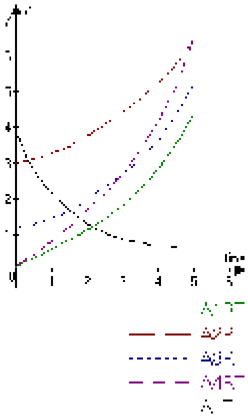

Figure 7: Scaled time series dataset AR with t_d= 5

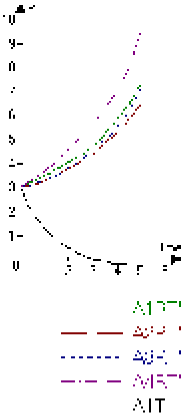

Figure 8: Scaled time series dataset AR with t_d = 5 and initial value of y = 3.0

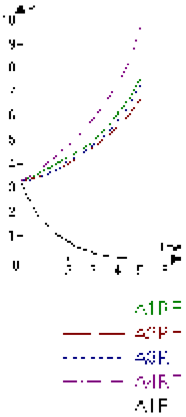

Figure 9: Final pre-processed time series dataset AR

Sequence	t_c	y_c
A1RF	3.73	11.73
A2RF	3.81	11.03
A3RF	3.77	11.47
A4RF	3.82	15.09
A1F	1.88	0.59

Table 3: Centroids for the time sequences AR

Time sequence pair	Distance between centroids	Percentage difference in the distance between centroids
(A1RF, A2R F)	0.71	173.08%
(A1RF, A3R F)	0.26	0%
(A1RF, A4R F)	3.36	1192.31%
(A1RF, A1F)	11.29	4242.31%

Table 4: Distance between centroids for time sequences in dataset AR

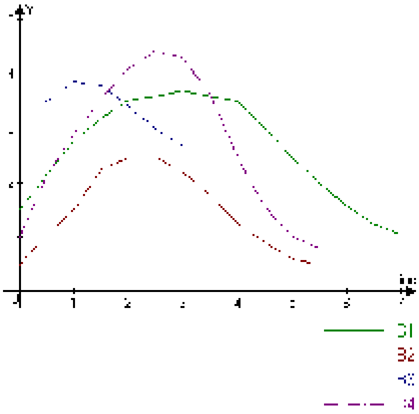

Figure 10: Time series dataset B

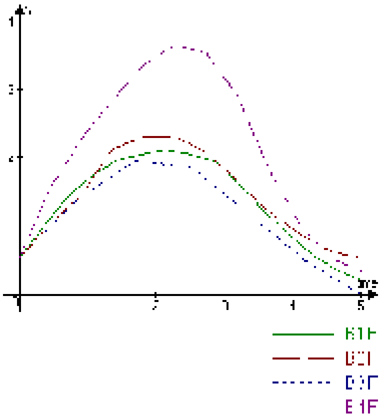

Figure 11: Final pre-processed time series dataset B

Sequence	t_c	y_c
B1F	2.82	1.98
B2F	2.96	2.12
B3F	2.65	1.78
B4F	2.81	3.42

Table 5: Centroids for the time sequences B

Time sequence pair	Distance between centroids	Percentage difference in the distance between centroids
(B1F, B2F)	0.19	0%
(B1F, B3F)	0.26	36.84%
(B1F, B4F)	1.44	657.89%

Table 6: Distance between centroids for time sequences in dataset B

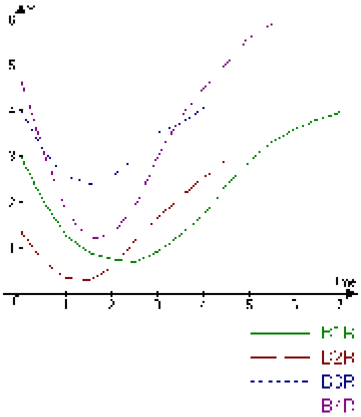

Figure 12: Time series dataset BR

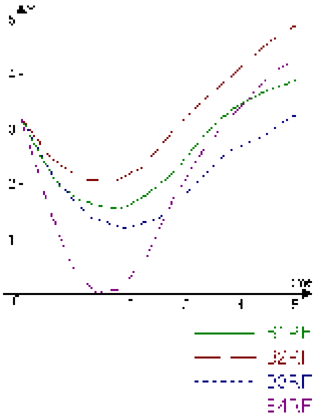

Figure 13: Final pre-processed time series dataset BR

Sequence	t_c	y_c
B1RF	3.78	6.39
B2RF	3.84	8.13
B3RF	3.89	5.39
B4RF	4.12	8.12

Table 7: Centroids for the time sequences BR

Time sequence pair	Distance between centroids	Percentage difference in the distance between centroids
(B1RF, B2RF)	1.74	72.28%
(B1RF, B3RF)	1.01	0%
(B1RF, B4RF)	1.79	77.23%

Table 8: Distance between centroids for time sequences in dataset BR

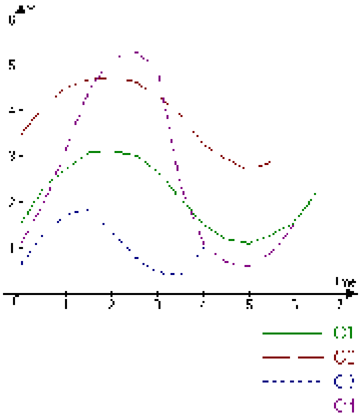

Figure 14: Time series dataset C

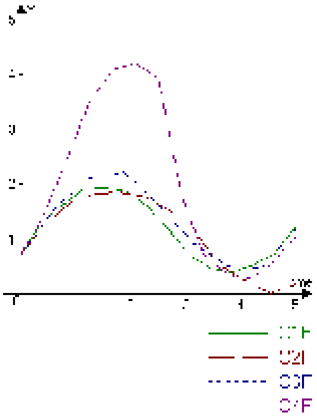

Figure 15: Final pre-processed time series dataset C

Sequence	t_c	y_c
C1F	3.08	1.67
C2F	2.48	1.51
C3F	3.11	1.89
C4F	2.63	3.11

Table 9: Centroids for the time sequences C

Time sequence pair	Distance between centroids	Percentage difference in the distance between centroids
(C1F, C2F)	0.62	181.82%
(C1F, C3F)	0.22	0%
(C1F, C4F)	1.51	586.36%

Table 10: Distance between centroids for time sequences in dataset B

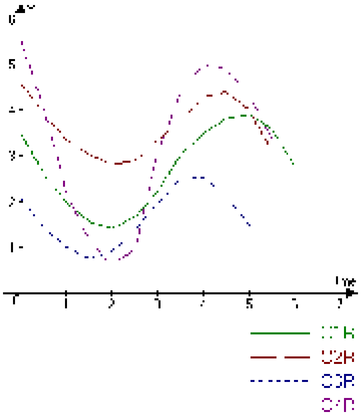

Figure 16: Time series dataset CR

The distance between the centroids of B1F and B2F is the minimum and that between B1F and B4F is the maximum. We may thus conclude that, in turn B1 (taken as query) is most similar to B2 and least similar to B4. The same conclusions can also be made by referring to Figure 11.

Figure 12 shows the time sequence dataset BR considered next. It is evident from Tables 7 and 8 that the centroid of B1RF (taken as query) is closest to that of B3RF and farthest from B4RF. Thus B1R (taken as query) is most similar to B3R and most dissimilar to B4R as is also evident from Figure 13.

In Figure 14 the time sequence dataset C has been shown. Its finally pre-processed version is shown in Figure 15. It is evident from Tables 9 and 10 that the centroid of C1F (taken as query) is closest to that of C3F and farthest from C4F. Thus C1 (taken as query) is most similar to C3 and most dissimilar to C4 as is also evident from Figure 15.

Figure 16 shows the time sequence dataset CR. Its finally pre-processed version is shown in Figure 17. It is evident from Tables 11 and 12 that the centroid of C1RF (taken as query) is closest to that of C3RF and farthest from C4RF.

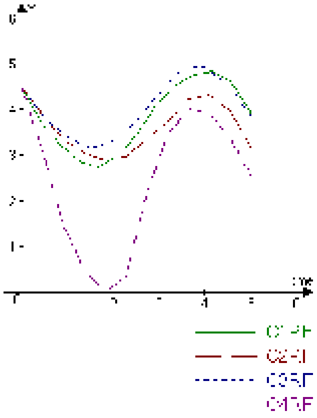

Figure 17: Final pre-processed time series dataset CR

Sequence	t_c	y_c
C1RF	3.61	7.81
C2RF	3.54	6.65
C3RF	3.59	7.84
C4RF	3.86	6.44

Table 11: Centroids for the time sequences CR

Time sequence pair	Distance between centroids	Percentage difference in the distance between centroids
(C1RF, C2RF)	1.16	2800.00%
(C1RF, C3RF)	0.04	0%
(C1RF,	1.39	3375.00%

C4RF)		

Table 12: Distance between centroids for time sequences in dataset CR

Thus it can be concluded that C1R (taken as query) is most similar to C3R and most dissimilar to C4R as is also evident from Figure 17.

5 Case Study

In this paper, we have taken stock movement data (Yahoo Finance) as our case data. The reason for choosing this data for our study is that stock movements have been successfully modelled as random walks (Agrawal, Faloutsos and Swami 1993). Random walk data has been used very commonly for similarity search in time series data (Agrawal, Faloutsos and Swami 1993).

The time series case data taken here consists of stock index movements for Dow Jones Industrial average (DJI), S & P 500 (SP), and NASDAQ Composite (NASDAQ) for all business days over a period of 4 months (last quarter) from September to December of each year from 1999 to 2003. Each of the time sequence consists of 85 datapoints. The index that has been studied is daily high. This data has been specifically studied keeping in view the effect of September 11, 2002 terrorist attacks in USA on movement of stock indices.

The data has first been pre-processed explained in Section 3. The transformed stock index data for daily high values in the last quarter of each year from 1999 to 2003 for NASDAQ is shown in Figures 18A and 18B respectively. The data for year 2003 has been taken as the query for similarity search and the others serve as candidates. The centroids have been shown in Table 13 and the distance between centroid of the query and candidate time sequences are shown in Table 14. It can be seen from Table 14 that the stock index movements in the last quarter of year 2003 is most similar to that for the year 2002 whereas it is most dissimilar to the index movement for the year 2000. The same can be concluded from Figures 18A and 18B.

The next set of data comprises of the variations of stock index (daily high) for DJI in the last quarter of each year from 1999 to 2003. The transformed data is shown in Figures 19A and 19B. The centroids are shown in Table 15 and the distance computations between centroids are shown in Table 16. It is evident from Table 16 that the stock index movement in the last quarter of year 2003 is the most similar to that for year 1999 and is most dissimilar to that for year 2000.

The same conclusions can be drawn after seeing Figures 19A and 19B.

The variations of stock index (daily high) for SP in the last quarter of each year from 1999 to 2003 have been considered next. The transformed data is shown in Figures 20A and 20B. The centroid computations are shown in Table 17 and the distance between centroids are computed in Table 18. It can be concluded from Table 18 that the stock index movements for last quarter of year 2003 is most similar to that for the year 1999 and most dissimilar to the year 2000. The same can be concluded from Figures 20A and 20B.

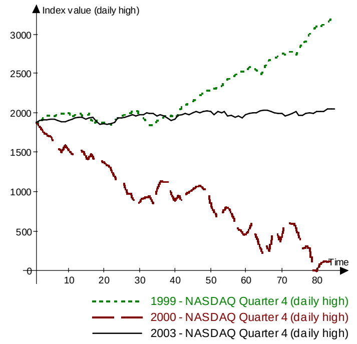

Figure 18A: Final pre-processed NASDAQ stock index data (daily high) from September to December

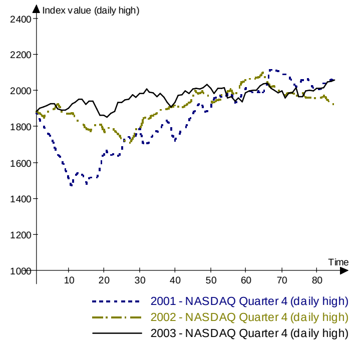

Figure 18B: Final pre-processed NASDAQ stock index data (daily high) from September to December

Sequence	t_c	y_c
1999	60.23	81415.36
2000	45.30	16650.06
2001	58.48	58184.32
2002	57.55	56757.38
2003(Query)	57.27	57166.98

Table 13: Centroids for time sequences indicating stock index movement for NASDAQ

Time sequence pair	Distance between centroids	Percentage difference in the distance between centroids
(1999, 2003)	24248.38	5820.00%
(2000, 2003)	40516.92	9791.82%
(2001, 2003)	1017.34	148.37%
(2002, 2003)	409.60	0%

Table 14: Distance between centroids

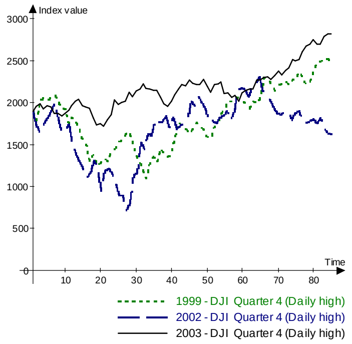

Figure 19A: Final pre-processed DJI stock index data (daily high) from September to December

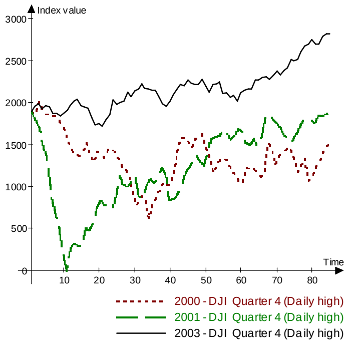

Figure 19B: Final pre-processed DJI stock index data (daily high) from September to December

Sequence	t_c	y_c
1999	60.43	64445.35
2000	57.17	37505.04
2001	61.91	50242.09
2002	58.72	54858.92
2003(Query)	59.01	70831.25

Table 15: Centroids for time sequences indicating stock index movement for DJI

Time sequence pair	Distance between centroids	Percentage difference in the distance between centroids
(1999, 2003)	6385.90	0%
(2000, 2003)	33326.21	421.87%
(2001, 2003)	20589.16	222.42%
(2002, 2003)	15972.33	150.12%

Table 16: Distance between centroids

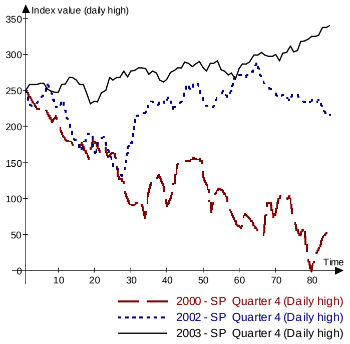

Figure 20A: Final pre-processed SP stock index data (daily high) from September to December

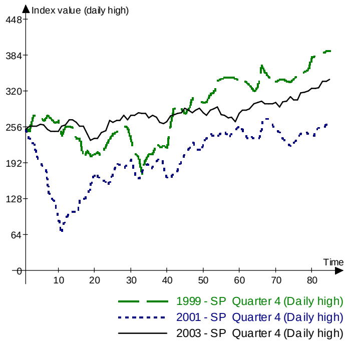

Figure 20B: Final pre-processed SP stock index data (daily high) from September to December

Sequence	t_c	y_c
1999	60.19	10212.26
2000	48.99	2450.23
2001	59.90	7123.23
2002	58.19	7053.12
2003(Query)	58.44	8794.77

Table 17: Centroids for time sequences indicating stock index movement for SP

Time sequence pair	Distance between centroids	Percentage difference in the distance between centroids
(1999, 2003)	1417.49	0%
(2000, 2003)	6344.55	347.59%
(2001, 2003)	1671.54	87.89%
(2002, 2003)	1741.65	22.87%

Table 18: Distance between centroids

6 Conclusions and Further Work

In this paper, a new and simple technique for performing similarity search in time series data using second moments (denoted as time weighted moments by us) has been proposed. We have assumed that the time series is comprised of a sequence of values representing a single measured variable. In the proposed approach, we apply a set of pre-processing steps to transform the given time series data and then calculate centroids using time weighted moments. The centroid points help us in making conclusions about the similarity of the time sequences. The proposed technique is capable of handling variable length queries and different baselines. It also works irrespective of global scaling of the data. The proposed approach does not involve any dimension reduction and hence the data distortions arising out of it are avoided. The paper also includes a case study on stock index movements.

For future work, we intend to extend our approach to multi-variable time series data in our approach. Also, we intend to develop a basket of parameters which may be used individually or in combination to assess similarity in time series data. One such parameter currently being studied by us is the variations in slopes of time sequences. Although, coincidental matches of the centroids is a remote possibility but it is intended to devise a basket of parameters which when used in conjunction to each other will completely rule out any such matches.

7 References

Guttman, A. (1994): R-trees: A dynamic index structure for spatial searching. *Proc. ACM SIGMOD International Conference on Management of Data*, 47-57.

Beckmann, N., Kriegel, H., Schneider, R. and Seeger, B. (1990): The R*-tree: An efficient and robust access method for points and rectangles. *Proc. ACM SIGMOD International Conference,* 322-331.

Kanth, K.V., Agrawal, D. and Singh, A. (1998): Dimensionality reduction for similarity searching in dynamic databases. *Proc. ACM SIGMOD International Conference,* 166-176.

Agrawal, R., Faloutsos, C. and Swami, A. (1993): Efficient similarity search in sequence databases. *Proc. 4th International Conference on Foundations of Data Organization and Algorithms*, Chicago, Illinois, USA, 69-84.

Kamel, I. and Faloutsos, C. (1994): Hilbert R-tree: An improved R-tree using fractals. *Proc. VLDB*, 500-509.

Chu, K. and Wong, M. (1999): Fast time-series searching with scaling and shifting. *Proc 18th ACM Symposium on Principles of Database Systems*, Philadelphia, PA, USA, 237-248.

Faloutsos, C., Jagadish, H., Mendelzon, A. and Milo, T. (1997): A signature technique for similarity based queries. *Proc. International Conference on Compression and Complexity of Sequences*, Positano-Salerno, Italy.

Refiei, D. (1999): On similarity based queries for time series data. *Proc. 15th IEEE International Conference on Data Engineering*, Sydney, Australia, 410-417.

Chan, K. and Fu , A. W. (1999): Efficient time series matching by wavelets. *Proc. 15th IEEE International Conference on Data Engineering*, Sydney, Australia, 126-133.

Kahveei , T. and Singh, A. (2001): Variable length queries for time series data. *Proc. 17th International Conference on Data Engineering*, Germany, 273-282.

Struzik, Z. and Siebes, A. (1999): The haar wavelet transform in the time series similarity paradigm. *Proc. Principles of Data Mining and Knowledge Discovery, 3rd European Conference*, Prague, Czech Republic, 12-22.

Korn, F., Jagadish, H. and Faloutsos, C. (1997): Efficiently supporting ad hoc queries in large datasets of time sequences. *Proc. ACM SIGMOD International Conference on Management of Data*, Tuescon, AZ, USA, 289-300.

Yi, Byoung-Kee and Faloutsos, C. (2000): Fast time sequence indexing for arbitrary Lp norms. *The VLDB Journal*, 385-394.

Faloutsos, C., Ranganathan, M. and Lopoulos, Y. M. (1994): Fast subsequence matching in time-series databases. *Proc. ACM SIGMOD International Conference on Management of Data*, 419-429.

Streeter, V. L. and Wylie, E. (1997): *Fluid Mechanics.* (Seventh Eds), McGraw-Hill International Book Company.

Yahoo Finance. http://finance.yahoo.com . Accessed 8 Nov 2004.

I have time series of parameters A, B, C and D. All of them are under influence of the same major conditions, but each one has minor differences. They are placed in different locations, A, B, C are in local1 and D is in local2. I would like to know which one (A, B, C) has the major similarity to D. How should I approach this issue? python classification data-mining time-series pandas. share | improve this question |. follow. | asked Dec 19 '16 at 11:22.