A Method of Similarity Measure and Visualization for Long Time Series Using Binary Patterns

Hailin Li[1], Chonghui Guo[1], and Libin Yang[2]

[1] Institute of Systems Engineering, Dalian University of Technology,
Dalian 116024, China
[2] College of Mathematics and Computer Science, Longyan University,
Longyan 364012, China
hailin@mail.dlut.edu.cn, guochonghui@tsinghua.org.cn,
ylib1982@163.com

Abstract. Similarity measure and visualization are two of the most interesting tasks in time series data mining and attract much attention in the last decade. Some representations have been proposed to reduce high dimensionality of time series and the corresponding distance functions have been used to measure their similarity. Moreover, visualization techniques are often based on such representations. One of the most popular time series visualization is time series bitmaps using chaos-game algorithm. In this paper, we propose an alternative version of the long time series bitmaps of which the number of the alphabets is not restricted to four. Simultaneously, the corresponding distance function is also proposed to measure the similarity between long time series. Our approach transforms long time series into SAX symbolic strings and constructs a non-sparse matrix which stores the frequency of binary patterns. The matrix can be used to calculate the similarity and visualize the long time series. The experiments demonstrate that our approach not only can measure the long time series as well as the "bag of pattern" (BOP), but also can obtain better visual effects of the long time series visualization than the chaos-game based time series bitmaps (CGB). Especially, the computation cost of pattern matrix construction in our approach is lower than that in CGB.

Keywords: Time series visualization, Binary patterns, Symbol representation, Similarity measure.

1 Introduction

Time series similarity measure is an interesting topic and also is a basic task in time series mining, which is an important tool for behavior informatics [3]. In the last decade, most of the studies have focused on the mining tasks based on similarity measure, including frequent patterns discovery, abnormal detection, classification, clustering, indexing and query. A common way to compare two

L. Cao et al. (Eds.): PAKDD 2011 Workshops, LNAI 7104, pp. 136–147, 2012.

time series is Euclidean distance measure. Since Euclidean distance treats time series elements independently and is sensitive to outliers [1], it is not suitable for calculating the distance between two long time series, let alone compare the time series whose length is different. Another popular method to compare time series is dynamic time warping (DTW)[2,4,5], which is "warping" time axis to make a good alignment between time series points. The distance of DTW is the minimum value and obtained by dynamic programming. DTW is more robust against noise than Euclidean distance and provides scaling along the time axis, but it is not suitable to measure the similarity of long time series because of its heavy computation cost, i.e. $O(mn)$, where m and n are the length of the two time series respectively.

Besides similarity measure in time series, another issue concerns the high dimensionality of time series. Two basic approaches are used to reduce the dimensionality, i.e. a piecewise discontinuous function and a low-order continuous function. The former mainly includes discrete wavelet transform (DWT)[6], piecewise linear approximation (PLA)[7], piecewise aggregate approximation (PAA) [8], symbolic aggregate approximation (SAX)[9,10] and their extended versions. The latter mainly includes non-linear regression, singular value decomposition (SVD) [11] and discrete fourier transforms (DFT)[12]. After dimensionality reduction by the above techniques, Euclidean distance can be used to measure the similarity of long time series.

Recently, SAX is a popular method used to represent time series, which transforms time series into a symbolic string and provides the corresponding distance function to measure the similarity, which is applied widely in many behavior-related cases. Especially, the bag-of-words representation [13] based SAX constructs the "bag of patterns" (BOP) matrix and uses it to measure the similarity of long time series mining including clustering, classification and anomaly detection. In addition, there are more and more visualization tools based on SAX appearing in the filed of time series data mining. Two of the most popular tools are the VizTree proposed by Lin [14,15] and chaos-game based time series bitmaps (CGB) [16]. They first convert each time series into a symbolic string with SAX, compute the frequency of the substrings (patterns) which are extracted from the string according to a sliding window with a fixed length, map the frequency to the color bar and finally construct the visualization with suffix tree [17] or quad-tree [16]. The visualizations entitles a user to discover clusters, anomalies and other regularities easily and fast. Especially, the chaos-game time series bitmaps (CGB) can arrange for the icons for time series files to appear as their bitmap representations. However, the number of alphabets used to divide normal distribution space in CGB is constricted to 4, which limits its applications. Moreover, the construction of pattern matrix demands a heavy computation cost. In most cases, the frequency matrix of patterns used to build the bitmaps is sparse [13], which means that little information on the time series is retained in the matrix and much extra memory space is used to store the data of which the value is 0. Furthermore, visual effects of time series bitmaps are not good for differentiating the long time series.

In this paper, we propose another alternative version to measure similarity of long time series and visualize them. It also transforms long time series into a symbolic string by SAX, easily constructs the matrix binary pattern which is a substring of length 2, calculates the frequency of each binary pattern, and form a binary pattern frequency matrix which can be used to measure the similarity and build the long time series bitmaps. The experiments demonstrate that our approach has the same clustering results as good as the method of "bag of patterns" (BOP), can also obtain non-sparse matrix of binary patterns and better visual effect with a lower time consumption of pattern matrix construction than that produced by CGB.

The rest of this paper is organized as follows. In section 2 we introduce the background and related work. In section 3 we propose our approach to measure similarity of long time series and visualize them. The empirical evaluation on clustering, visualization and computation cost comparison on long time series dataset is presented in section 4. In the last section we conclude our work.

2 Background and Related Work

Symbolic aggregate approximation (SAX) is a good method to simply represent time series. Time series is transformed into a symbolic string by SAX and the string can be used to calculate the similarity between two time series. Furthermore, the string also can be applied to draw bitmaps for the visualization of time series.

SAX represents time series by some fixed alphabets (words). It initially normalizes time series into a new sequence with the mean of 0 and the variance of 1, and then divides it into equal-sized w sections. The mean of each section is calculated and further represented by an alphabet whose precinct includes the mean. Fig. 1 shows one time series transformed into one string "DACCADADBB" by SAX.

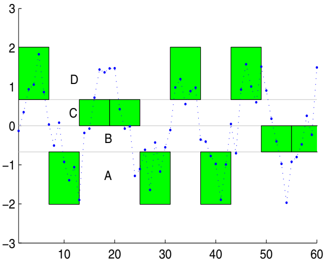


Fig. 1. Time series is transform into a string "DACCADADBB" by SAX

SAX representation can be used to find structural similarity in long time series. The paper [13] presented a histogram-based representation for the long time series, called it "bag of patterns" (BOP), presented in paper [13]. The word-sequence (pattern) matrix can be constructed after obtaining a set of strings for each time series. The matrix is used to calculate the similarity by any applicable distance measure or dimensionality reduction techniques. This approach is widely accepted by the text mining and information retrieval communities.

Furthermore, SAX representation can also be used to create the time series bitmaps. The authors [16] let SAX representation form a time series bitmaps by the algorithm of "chaos game" [18], which can produce a representation of DNA sequences. We call this Chaos-Game based time series Bitmaps CGB for short. CGB are defined for sequences with an alphabet size of 4. The four possible SAX symbolics are mapped to four quadrants of a square and each quadrant can be recursively substituted by the organized pattern in the next level. The method computes the frequencies of the patterns in the final square, linearly maps those frequencies to colors, and finally constructs a square time series bitmaps. Fig. 2(a) illustrates this process. The detail algorithm can be referred to the paper [16].

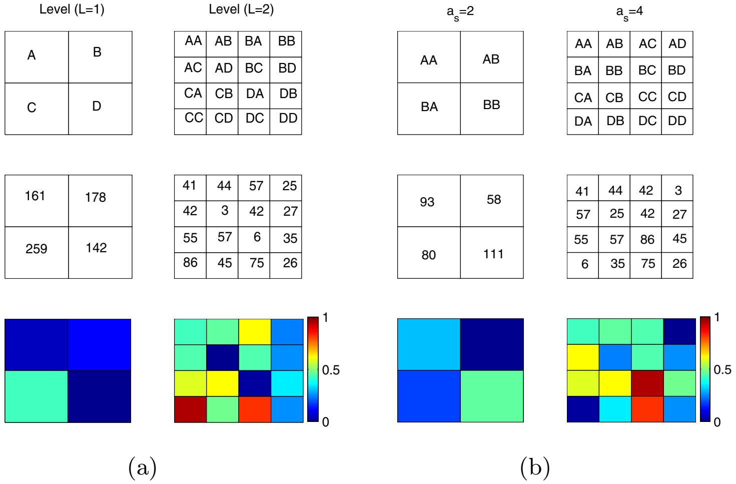


Fig. 2. (a) The process of time series bitmap construction by CGB. (b) The process of time series bitmap construction by our approach using binary patterns.

3 Binary Patterns Based Similarity and Visualization

Although BOP [13] is superior to many existing methods to measure similarity of long time series in the tasks of clustering, classification and anomaly detection, one of the disadvantages is that the size of possible SAX strings (the resulting dictionary size) is very large, which causes the word-sequence (pattern) matrix be sparse so that only few information on the original time series is retained. For example, with $\alpha_s = 4$ and $w = 10$, the size of the possible SAX strings is $\alpha_s^w = 4^{10} = 1048576$. Moreover, such large size needs a great of the memory space to store the data including many data points of which the value is 0. Even if we use a compress format [19] to store the data by compress column storage, the extra computation cost is likely to increase.

The chaos-game based time series bitmaps (CGB) [16] are only constructed by four symbolics, which is often used in DNA analysis. However, its applications are confined by the four symbolics because different number of the symbolics used to mine time series is demanded in most cases. With the high level in the square, the possibility of the frequent patterns in the long time series is quite low, which affects the visual effect of time series bitmaps. Moreover, it also needs a great of time consumption to construct the pattern matrix because of the recursive operation. Therefore, it is not reasonable to construct the bitmaps for long time series in the high level.

In this paper, we propose an alternative method to measure similarity and construct the bitmaps for long time series. It at least can overcome the above mentioned problems. For convenience, we give several definitions used to design our approach for long time series.

Definition 1. Alphabet set. The normalized time series is subject to the standard normal distribution $N(0, 1)$. Alphabet set $S = \{s_1, s_2, \dots, s_{a_s}\}$ is used to equiprobably divide the distribution space into a_s regions.

If the position of one point in time series locates the *ith* region which is represented by a alphabet s_i, then the point can be denoted as s_i.

Definition 2. The symbolic string of time series. A symbolic sequence $Q' = \{q'_1, q'_2, \dots, q'_w\}$ is obtained from a time series $Q = \{q_1, q_2, \dots, q_m\}$ by SAX.

Definition 3. Binary pattern. If the length of a substring is equal to 2, we call the substring a binary pattern.

For example, if a string obtained by SAX is $BAABCAC$, then the set of the binary patterns is $\{BA, AA, AB, BC, CA, AC\}$.

Definition 4. Binary pattern matrix (BPM). The alphabet set S is used to represent the time series, then the binary pattern matrix can be defined as

$$\begin{pmatrix} s_1s_1 & s_1s_2 & \dots & s_1s_{a_s} \\ s_2s_1 & s_2s_2 & \dots & s_2s_{a_s} \\ \vdots & \vdots & \vdots & \vdots \\ s_{a_s}s_1 & s_{a_s}s_2 & \dots & s_{a_s}s_{a_s} \end{pmatrix}.$$

For example, if the alphabets $S = \{A, B, C\}$, then the binary pattern matrix is

$$\begin{pmatrix} AA & AB & AC \\ BA & BB & BC \\ CA & CB & CC \end{pmatrix}.$$

Definition 5. Binary pattern frequency matrix (BM). Suppose there is a string and the binary pattern matrix, the frequency of the binary pattern in the string can be computed. Those frequencies of the binary patterns constitute the Binary frequency matrix.

For example, if there is a string *ababaabccbacbaa*, then the binary pattern frequency matrix (BM) is

$$\begin{pmatrix} 2 & 3 & 1 \\ 4 & 0 & 1 \\ 0 & 2 & 1 \end{pmatrix}.$$

A long time series Q, of which the length is m, is transformed into $m - N + 1$ strings according to a sliding window of which the length is N by SAX. Each string produces a binary pattern frequency matrix BM_i, where $1 \leq i \leq m-N+1$. we can sum all the BM_i to obtain the total BM for the long time series, i.e. $BM = BM_1 + BM_2 + \ldots + BM_{m-N+1}$.

The inspired mind of our approach is something like the BOF method, which uses the word-sequence matrix to measure the similarity of long time series. However, it is obvious that the matrix of our approach is not sparse even though the alphabet size a_s is large. Moreover, since the number of binary patterns is fewer than that of the patterns produced by BOF and CGB in the high level, the memory used to store the binary patterns is lower.

Since the size of binary pattern matrix is small, the Euclidean distance function is a good choice to measure the similarity, i.e.

$$D(Q, C) = \sqrt{\sum_{j=1}^{a_s}\sum_{i=1}^{a_s}(BM_Q(i,j) - BM_C(i,j))^2}, \tag{1}$$

where BM_Q and BM_C are the binary frequency matrix of the long time series Q and C respectively.

After obtaining the BM, we also can apply it to construct the time series bitmaps. We can normalize the elements of BM and let every element's value locate into $[0, 1]$. The normalization function is not unique and can be formed by various ways. In our approach, the normalization function is

$$BFM = \frac{BM - min(BM)}{max(BM) - min(BM)} \in [0, 1], \tag{2}$$

where $min(BM)$ and $max(BM)$ can return the minimum and the maximum of the values in the BM.

Each normalized element in the matrix can be mapped to the color of which the value also locates in [0,1]. In this way, the matrix can be transform a bitmaps as shown in Fig. 2(b). We need to point out that the Matlab color bar is provided to construct all time series bitmaps in this paper. We know that when the alphabet size a_s in our approach is equal to 4 and the level L of quad-tree in the CGB approach is equal to 2, i.e. $a_s = 4$ and $L = 2$, both of the approaches have the same elements of sequence matrix so that their corresponding frequencies and the colors of the binary patterns are also equal as shown in Fig. 2(a) and 2(b). For example, The frequency of the binary pattern AB is equal to 41 in both approaches and their colors are also identical when there is $a_s = 4$ and

$L = 2$. It demonstrates that the proposed method is available to measure and visualize time series as well as the traditional approaches.

For larger value of the parameter in the two respective methods, a_s in our method and L in the CGB approach, the number of binary patterns produced by our method is much less than that of other non-binary patterns produced by CGB and the binary patterns in our method can provide more sufficient information to approximate the long time series. Our approach using the matrix BM not only can conveniently measure the similarity of the long time series but also can construct the time series bitmaps with a good visual effect. Moreover, the computation cost for the pattern matrix construction is lower than the CGB. All those information can be demonstrated in next section. It is necessary to point out that, unlike the CGB method, the length of patterns in our approach is equal to 2, it can't deal with the patterns with different length except for the length 2. That is the reason why we call our approach an alternative version of long time series bitmaps.

4 Empirical Evaluation

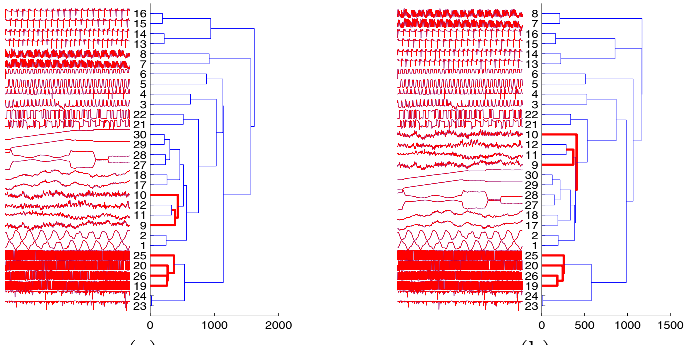
In this section, we perform the hierarchical clustering to demonstrate that our approach can produce the same result as BOP [16] does, which also means that our approach is available for time series mining as good as BOP. We also compare the visual effects between our approach and CGB in the setting of different parameters to show that our approach has higher quality of visual effect of the long time series. The computation cost comparison between our approach and CGB for the pattern matrix construction is also discussed in the last subsection.

4.1 Hierarchical Clustering

It is well known that hierarchical clustering is one of the most widely used approaches to produce a nested hierarchy of the clusters. It can offer a great visualization power to analyze the relationships among the different time series. Moreover, unlike other clustering method, it does not require any input parameters.

We experiment on the long time series dataset derived from the paper [16] , of which the length is 1000. Fig. 3(a) and Fig. 3(b) show the resulting dendrogram for our approach and BOP when the size of the binary pattern matrix in our approach is equal to that of the word-sequence matrix in the BOP, i.e. 8×8. we can find that our method has the same clustering results as BOP does. Especially, when the alphabets size a_s is equal to 4 for our approach and the level is equal to 2, then the clustering results of two approach are identical as shown in Fig. 4. The reason is that in that case the binary pattern matrix (BPM) of our approach is corresponding to the word-sequence matrix of BOP.

From the clustering results we know that the performance of similarity measure in our approach is same to BOP. It also means that our approach is available for long time series data mining as BOP had done.



(a) (b)

Fig. 3. (a) The clustering result of our approach is obtained when the alphabet size is 8 and , i.e. $(a_s = 8, N = 100, w = 10)$. (b) The clustering result of BOP is obtained when the level of quad-tree is 3, i.e. $(L = 3, N = 100, w = 10)$. N is the length of sliding window and w is the length of substring. Bold red lines denote incorrect substree.

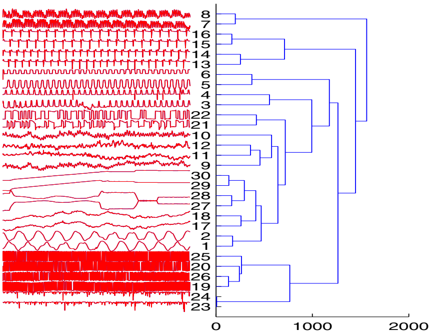


Fig. 4. The identical clustering result is obtained for our approach and BOP when $a_s = 4$ and $L = 2$ for the two approaches and $(N = 100, w = 10)$

4.2 Visual Effects

We also use the previous 30 long time series to produce the bitmaps by our approach and CGB. We compare their visual effects under the same size of the binary pattern matrix and word-sequence matrix. We provide two groups to make the experiments. One is under the matrix size of 4×4, that is, the alphabet size of our approach is 4 (i.e. $a_s = 4$) and the level of the quad-tree is 2 (i.e. $L = 2$). The results of the two approaches are shown in Fig. 5(a) and Fig. 5(b) respectively. The other is under the matrix size of 8×8, that is, the alphabet size of our approach is 8 (i.e. $a_s = 8$) and the level of the quad-tree is 3 (i.e. $L = 3$). The results of the two approaches are shown in Fig. 6(a) and Fig. 6(b) respectively.

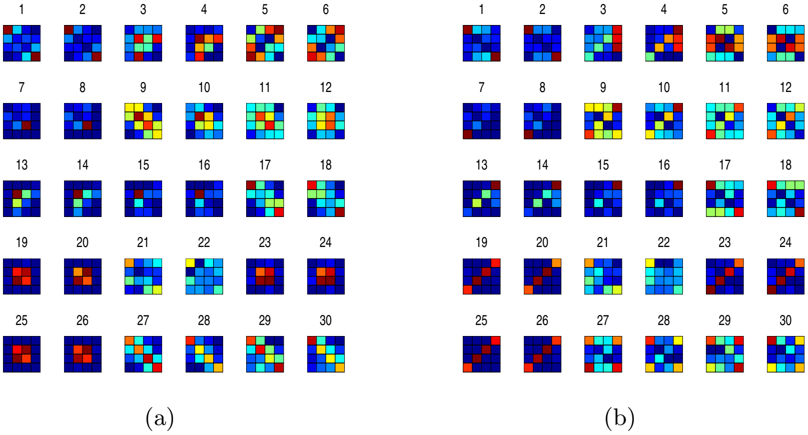


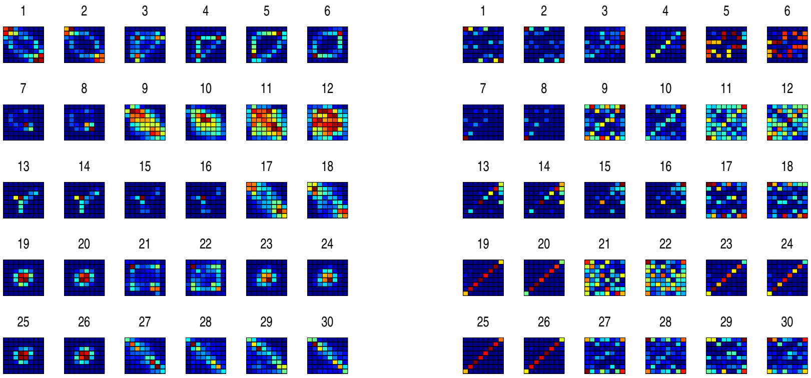
Fig. 5. (a) The bitmaps are constructed by our approach under $a_s = 4$. (b) The bitmaps are constructed by the CGB under $L = 2$.

It is easy to find that in Fig. 5(a) the color of each grid of one bitmap can be mapped to the color of each grid of the corresponding bitmap in Fig. 5(b) . As shown in Fig. 7(a) , the color of the grid is mapped to each other in that case, i.e. $a_s = 4$ in our approach and $L = 2$ in CGB. In other word, our alternative method to visualize long time series is available as CGB does.

In Fig. 6(a) and Fig. 6(b), it is obvious that the visual effect of our approach is better than the CGB. The hierarchical clustering results in previous subsection tell us that the group of time series 11 and 12 and the group of time series 21 and 22 are in the different clusters. However, Fig. 6(b) produced by CGB regard the four bitmaps as the members of the same cluster. But in Fig. 6(a) produced by our approach it is easy to distinguish the two groups. Other bitmaps of long time series also have such cases. Therefore, the visual effects of long time series bitmaps produced by our approach are better than those produced by CGB.

4.3 Comparison of Computation Cost

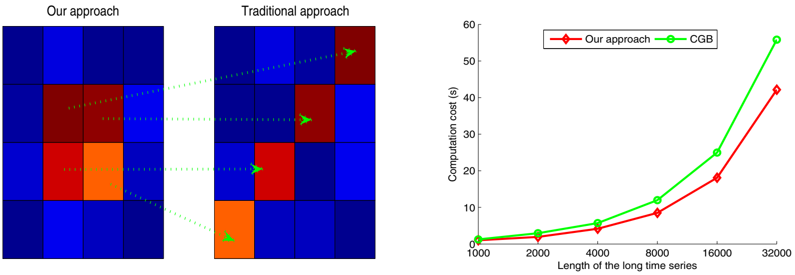
Since the computation cost of the pattern matrix construction is one of the most important differentia between our approach and the CGB, we make the experiments about the computation cost of the pattern matrix construction 6 times. Every time we truncate 50 time series with the same length from the long stock time series [20]. The average result of the computation cost is shown in Fig. 7(b) when the alphabet size a_s in our approach is equal to 4 and the quad-tree level in CGB is equal to 2. In other word, the comparison of the time computation is carried out under the same size of the pattern matrix for the two methods. It is obvious that the time consumption in our approach is lower than that in CBG.



(a)

(b)

Fig. 6. (a) The bitmaps are constructed by our approach under $a_s = 8$. (b) The bitmaps are constructed by the CGB under $L = 3$.



(a)

(b)

Fig. 7. (a) The color of the grid is mapped to each other under the matrix size of 4 × 4. (b)The time consumption of pattern matrix construction is compared between our approach and the CGB.

5 Conclusions

In this work, we have proposed an alterative approach to measure the similarity measure and visualize the long time series, which is based on binary patterns. Our approach counts the frequency of the occurrences of each binary patterns and forms a binary pattern frequency matrix. According to this matrix, we can measure the similarity of the long time series by Euclidean distance function and construct the bitmaps for the visualization of the long time series. In the experiments we demonstrated the effectiveness of our approach on the hierarchical clustering for the long time series, which shows that our approach can measure the similarity well as BOP does. Furthermore, the visual effect of time series bitmaps produced by our approach is better than the traditional one (CGB). From the comparison of the computation cost of pattern matrix construction, we conclude that our approach is also more efficient.

Acknowledgments. This work has been partly supported by the Natural Science Foundation of China under Grant Nos. 70871015 and 71031002.

References

1. Agrawal, R., Lin, K.I., Sawhney, H.S., Shim, K.: Fast similarity search in the presence of noise, scaling and translation in time-series databases. In: Proceedings of Very Large DataBase (VLDB), pp. 490–501 (1995)
2. Berndt, D.J., Clifford, J.: Finding patterns in time series: A dynmaic programming approach. In: Advances in Knowledge Discovery and Data Mining, pp. 229–248 (1996)
3. Cao, L.: In-depth Behavior Understanding and Use: the Behavior Informatics Approach. Information Science 180(17), 3067–3085 (2010)
4. Rabiner, L., Juang, B.H.: Fundamentals of speech recognition, Englewood Cliffs, N.J (1993)
5. Keogh, E.: Exact indexing of dynamic time warping. In: Proceedings of the 28th VLDB Conference, Hong Kong, China, pp. 1–12 (2002)
6. Popivanov, I., Miller, R.J.: Similarity search over time-series data using wavelets. In: Proceedings of the 18th International Conference on Data Engineering, pp. 212–221 (2002)
7. Iyer, M.A., Harris, M.M., Watson, L.T., Berry, M.W.: A performance comparison of piecewise linear estimation methods. In: Proceedings of the 2008 Spring Simulation Multi-Conference, pp. 273–278 (2008)
8. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing SAX: a novel symbolic representation of time series. Data Mining and Knowledge Discovery 15, 107–144 (2007)
9. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pp. 2–11 (2003)
10. Keogh, E., Lin, J., Fu, A.: Hot SAX: efficiently finding the most unusual time series subsequence. In: Proceedings of the 5th IEEE International Conference on Data Mining, pp. 226–233 (2005)
11. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 4th edn., pp. 323–409 (2009)
12. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time series databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 419–429 (1994)
13. Lin, J., Li, Y.: Finding Structural Similarity in Time Series Data using Bag of Patterns Representation. In: Winslett, M. (ed.) SSDBM 2009. LNCS, vol. 5566, pp. 461–477. Springer, Heidelberg (2009)
14. Lin, J., Keogh, E., et al.: VizTree: a tool for visually mining and monitoring massive time series databases. In: Proceedings 2004 VLDB Conference, pp. 1269–1272. Morgan Kaufmann, St Louis (2004)
15. Lin, J., Keogh, E., et al.: Visually mining and monitoring massive time series. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, pp. 460–469 (2004)
16. Kumar, N., Lolla, V.N., et al.: Time-series bitmaps: a practical visualization tool for working with large time series databases. In: SIAM 2005 Data Mining Conference, pp. 531–535 (2005)

17. Fu, T.C., Chung, F.L., Kwok, K., Ng, C.M.: Stock time series visualization based on data point importance. Engineering Applications of Artificial Intelligence 21(8), 1217–1232 (2008)
18. Barnsley, M.F.: Fractals everywhere, 2nd edn. Academic Press (1993)
19. Ekambaram, A., Montagne, E.: An Alternative Compress Storage Format for Sparse Matrices. In: Yazıcı, A., Şener, C. (eds.) ISCIS 2003. LNCS, vol. 2869, pp. 196–203. Springer, Heidelberg (2003)
20. Stock.: Stock data web page (2005), `http://www.cs.ucr.edu/~wli/FilteringData/stock.zip`