社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2004－ICS－138 (26)
2004／12／5

Clustering Time-series Data Based on the Modified Multiscale Matching Technique

Shoji Hirano† and Shusaku Tsumoto†

This paper presents an improved version of time-series multiscale matching method that eludes the problem of shrinkage. The key idea is the development of new segment representation. The shape parameters of a segment at high scale are now directly obtained using the shape parameters of base segments at the lowest scale, instead of using shapes represented by multiscale description. Multiscale shapes are now used only to obtain the hierarchy of the segments; since segment parameters are obtained independently of multiscale shapes, shrinkage does not distort them. We examined the usefulness of the method on the cylinder-bell-funnel dataset. The results demonstrated that the dissimilarity matrix produced by the proposed method, combined with conventional clustering techniques, lead to the successful clustering.

1. Introduction

Clustering of time-series data provides an efficient way of finding the groups of sequences with respect to their similarity of temporal courses, as well as a way of revealing the underlying structure of the dataset[1]. One of the difficulties in time-series clustering is the comparison of two time series, in the light that (1)the length of series can be different, and (2)the series may represent *partly* similar structures. Different length of time series induces the requirement of one-to-many matching of temporal data points, involving the new problem of finding the best match. Generally, a time-series can be regarded as a high-dimensional data where one temporal data corresponds to one dimension, meaning that computational costs for comparing each data value is not negligible. Therefore, a simple and efficient comparison methods such as Dynamic Time Warping[2),3)], or frequency domain based approaches[4] are often used for comparing time series of different length. However, since these methods do not directly take into account the structural similarity of time series, the results may not reflect the local and global similarities of temporal events.

As a structural comparison method of time series, we have developed a method based on multiscale matching[5]. Multiscale representation/matching[6),7)], developed originally as pattern recognition methods in computer vision, have an ability to compare two shapes by partly changing observation scales. We have extended their approach to deal with multiscale, structural comparison of time series, including redesign of dissimilarity measures and matching scheme. However, it still inherits a problem called shrinkage, an excessive distortion of shapes at high scales, which makes shape-based dissimilarity be also largely distorted.

This paper presents an improved version of time-series multiscale matching method that eludes the problem of shrinkage. The key idea is the development of new segment representation. The shape parameters of a segment at high scale are now directly obtained using the shape parameters of base segments at the lowest scale, instead of using shapes represented by multiscale description. Multiscale shapes are now used only to obtain the hierarchy of the segments; since segment parameters are obtained independently of multiscale shapes, shrinkage does not distort them. Besides, we introduce an alternative smoothing kernel composed of the modified Bessel function, so that the causality in scale dimension can be held also with discrete signals, and shapes at very low scale can be represented properly. We examined the usefulness of the method on the cylinder-bell-funnel dataset. The results demonstrated that the dissimilarity matrix produced by the proposed method, combined with conventional clustering techniques, lead to the successful clustering.

† Department of Medical Informatics,
Shimane University, School of Medicine
89-1, Enya-cho, Izumo, Shimane 693-8501 Japan

2. Basics of Time Series Multiscale Matching and Problem

Multiscale matching representation/matching[6),7)] is originally developed as a method for comparing two planar curves by partly changing observation scales. It divides a contour of the object into partial contours based on the place of inflection points. After generating partial contours at various scales for each of the two curves to be compared, it finds the best pairs of partial contours that minimize the total dissimilarity while preserving completeness of the concatenated contours. This method can preserve connectivity of partial contours by tracing hierarchical structure of inflection points on the scale space. Since each ends of a partial contour exactly corresponds to an inflection point and the correspondence between inflection points at different scales are recognized, the connectivity of the partial contours is guaranteed. We have extended this method so that it can be applied to the comparison of two one-dimensional temporal sequences. A planar curve can be redefined as a temporal sequence, and a partial contour can be analogously redefined as a subsequence.

Now let us introduce the basics of multiscale matching for one-dimensional temporal sequence. First, we represent time-series A using multiscale description. Let $x(t)$ represent an original temporal sequence of A where t denotes a time of data acquisition. The sequence at scale σ, $X(t,s)$, can be represented as a convolution of $x(t)$ and a Gauss function with scale factor σ, $g(t,\sigma)$, as follows:

$$\begin{aligned} X(t,\sigma) &= x(t) \otimes g(t,\sigma) \\ &= \int_{-\infty}^{+\infty} x(u) \frac{1}{\sigma\sqrt{2\pi}} e^{-(t-u)^2/2\sigma^2} du. \end{aligned}$$

A sequence will be smoothed at higher scale and the number of inflection points is also reduced at higher scale. Curvature of the sequence can be calculated as

$$K(t,\sigma) = \frac{X''}{(1+X'^2)^{3/2}},$$

where X' and X'' denotes the first- and second-order derivatives of $X(t,\sigma)$, respectively. The m-th derivative of $X(t,\sigma)$, $X^{(m)}(t,\sigma)$, is derived as a convolution of $x(t)$ and the m-th order derivative of $g(t,\sigma)$, $g^{(m)}(t,\sigma)$, as

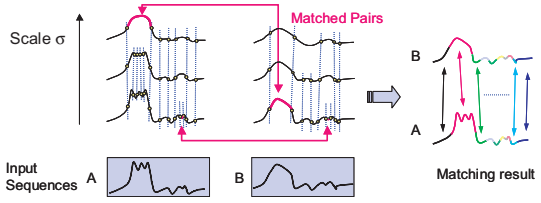

Fig. 1 Multiscale matching.

$$X^{(m)}(t,\sigma) = \frac{\partial^m X(t,\sigma)}{\partial t^m} = x(t) \otimes g^{(m)}(t,\sigma).$$

The next step is to find inflection points according to change of the sign of the curvature and to construct segments. A segment is a subsequence whose ends respectively correspond to the adjacent inflection points. $\mathbf{A}^{(k)}$ be a set of N segments that represents the sequence at scale $\sigma^{(k)}$. $\mathbf{A}^{(k)}$ can be represented as

$$\mathbf{A}^{(k)} = \left\{ a_i^{(k)} \mid i = 1, 2, \cdots, N^{(k)} \right\}.$$

In the same way, for another temporal sequence B, we can obtain a set of segments $\mathbf{B}^{(h)}$ at scale $\sigma^{(h)}$ as

$$\mathbf{B}^{(h)} = \left\{ b_j^{(h)} \mid j = 1, 2, \cdots, M^{(h)} \right\},$$

where M denotes the number of segments of B at scale $\sigma^{(h)}$.

The main procedure of multiscale structure matching is to find the best set of segment pairs that minimizes the total difference. Figure 1 illustrates the process. For example, five contiguous segments at the lowest scale of Sequence A are integrated into one segment at the highest scale, and the integrated segments well match to one segment in Sequence B at the lowest scale. Thus the set of the five segments in Sequence A and the one segment in Sequence B will be considered as a candidate for corresponding subsequences. While, another pair of segments will be matched at the lowest scale. In this way, matching is performed throughout all scales. The resultant set of segment pairs must not be redundant or insufficient to represent the original sequences. Namely, by concatenating all segments in the set, the original sequence must be completely reconstructed without any partial gaps or overlaps. The matching process can be fasten by implementing dynamic programming scheme[7)].

Definition of segment difference is a key factor in multiscale matching. As Lowe reported[8)], sequences at high scales involve the problem of

shrinkage, because convolution with Gaussian function at high scales results in excessive averaging with too wide neighbors. It means that a sequence will be largely distorted from the original one at high scales; the shape of the sequence will become close to line (1-dimension case) or circle(2-dimension case). Therefore, it should be avoided to define the segment difference directly using segment shapes at high scales. In order not to use higher scale shapes, Ueda et al.[7] added a cost factor that suppresses excessive replacement of segments into their segment difference:

$$d(a_i^{(k)}, b_j^{(h)}) = d_{angle}(a_i^{(k)}, b_j^{(h)}) \times d_{length}(a_i^{(k)}, b_j^{(h)}) + \gamma(\mathrm{cost}(a_i^{(k)}) + \mathrm{cost}(b_j^{(h)}))$$

where d_{angle} and d_{length} respectively represent the difference of rotation angle and relative length of segments defined below.

$$d_{angle}(a_i^{(k)}, b_j^{(h)}) = \frac{\mid \theta_{a_i}^{(k)} - \theta_{b_j}^{(h)} \mid}{\theta_{a_i}^{(k)} + \theta_{b_j}^{(h)}}$$

$$d_{length}(a_i^{(k)}, b_j^{(h)}) = \left| \frac{l_{a_i}^{(k)}}{L_A^{(k)}} - \frac{l_{b_j}^{(h)}}{L_B^{(h)}} \right|$$

where $\theta_{a_i}^{(k)}$ and $\theta_{b_j}^{(h)}$ denote rotation angles of tangent vectors along segments ${a_i}^{(k)}$ and ${b_j}^{(h)}$, $l_{a_i}^{(k)}$ and $l_{b_j}^{(h)}$ denote the length of segments, $L_A^{(k)}$ and $L_B^{(h)}$ denote the total length of sequences$A$ and B at scales $\sigma^{(k)}$ and $\sigma^{(h)}$, respectively. $\mathrm{Cost}(a_i^{(k)})$ is a cost to form $a_i^{(k)}$. If $a_i^{(k)}$ is formed by merging some segments on the lowest scale, appropriate cost is added. A constant γ is a weight for cost.

However, this approach involves a problem that, it is difficult to correctly evaluate the global similarity of sequences, because of the following reasons.

- Dissimilarity components are calculated directly using shapes at high scales, which are excessively distorted by the convolution with Gaussian kernel. As we described previously, every segment will be close to a line at a enough high scale; therefore, the difference of rotation angle will be close to zero, regardless of how their base segments are different.
- In order to prevent to such an excessive distortion, a cost factor defined as a sum of segment differences of merged segments is added. However, at high scales, dissimilarity $d(a_i^{(k)}, b_j^{(h)})$ is mostly occupied by the cost factor; which does not surely represent the dissimilarity of segments $a_i^{(k)}$ and $b_j^{(h)}$.

3. New segment representation

Based on the above observation, we have designed a new multiscale matching scheme for temporal sequences. The key points are:

- Multiscale shapes are used only to obtain the hierarchy of the segments.
- Segment difference and replacement cost are directly obtained from segments at the lowest scale.

Figure 2 left provides our shape parameters for single segment a_i. Since we use only segments at the lowest scale, we omit (0) of $a_i^{(0)}$ for simplicity. Features of a_i consist of the following four components:

(1) Amplitude $\mathrm{a}(a_i)$; vertical amplitude measured from the baseline of the segment to peak point $\mathrm{pk}[a_i]$. Baseline is a straight line connecting both ends of the segment.
(2) Width $\mathrm{w}(a_i)$; horizontal width of the segment.
(3) Height $\mathrm{h}(a_i)$; vertical shift of the segment w.r.t. both ends.
(4) Phase $\mathrm{p}(a_i)$; phase of the segment; measured from the starting point of the entire sequence to the starting point of the segment.

Figure 2 center illustrates our shape parameters for merged segments. Suppose that n contiguous segments $(a_i, a_{i+1}, \ldots, a_{i+n-1})$ can be replaced into one segment $a_m^{(k)}$ at scale k according to the segment hierarchy. Then we generate the new shape parameters for the replaced segment $a_m^{(k)}$ as follows: (1) determine the new peak point to the centroid $pk[a_i, a_{i+1}, \ldots, a_{i+n-1}]$ of all n peaks. (2) using $pk[a_i, a_{i+1}, \ldots, a_{i+n-1}]$ and both ends of the merged segment, calculate the four parameters $a(a_m^{(k)})$, $w(a_m^{(k)})$, $h(a_m^{(k)})$, $p(a_m^{(k)})$. Note that we do not directly use the shape of in order to obtain the above four parameters. Figure 2 right illustrates the problem of shrinkage. Since we have multiscale description, we can obtain the shape of $a_m^{(k)}$ at scale k and directly calculate

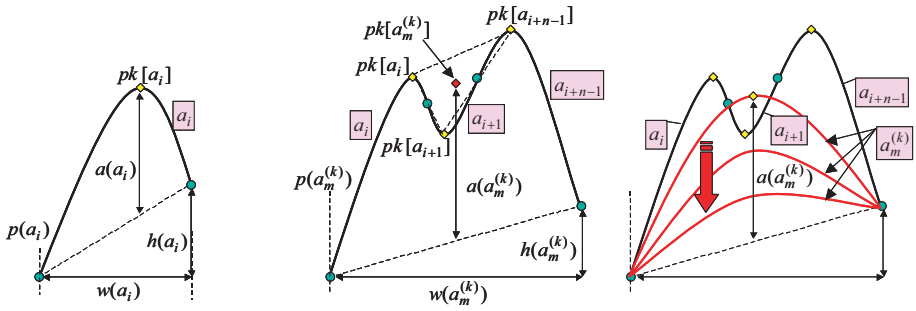

Fig. 2 Illustrative examples of new segment parameters.

$a(a_m^{(k)})$ as a single segment. However, it will shrink toward the baseline as k becomes large, distorting the amplitude value (actually, all of the four parameters are subject to be distorted by shrinkage).

Now the segment difference is defined as:

$$d(a_m^{(k)}, b_l^{(h)}) = \max(d_a, d_w, d_h, d_p) + \gamma(\mathrm{cost}(a_m^{(k)}) + \mathrm{cost}(b_l^{(h)}))$$

where $d_a = d_a(a_m^{(k)}, b_l^{(h)})$ denotes difference of amplitude between segments $a_m^{(k)}$ and $b_l^{(h)}$, and other symbols d_w, d_h, d_p are interpreted analogously. $\mathrm{Cost}(a_m^{(k)})$ is a cost function for replacement. For $a_m^{(k)}$ that replaces $(a_i, a_{i+1}, \ldots, a_{i+n-1})$, cost is defined by

$$\mathrm{cost}(a_m^{(k)}) = \begin{cases} 0, & \text{if } n = 1 \\ \min\left(\left\{\frac{\frac{1}{n}\sum_{u=i}^{i+n-1}|a(a_u^{(0)})|}{a(a_m^{(k)})}\right\}, 1\right) & \text{otherwise.} \end{cases}$$

This cost becomes low when the replacement enlarges average amplitude.

Based on the above dissimilarity measure $d(a_m^{(k)}, b_l^{(h)})$, the best set of segment pairs $\mathbf{P} = \{p_w, 1 \le w \le N_p \mid p_1 = ((a_1 - a_i, b_1 - b_l)), p_2 = (a_{i+1} - a_j, b_{l+1} - bm), \ldots, p_{Np} = (a_{k+1} - a_N, b_{n+1} - b_M)\}$, that minimizes the accumulated difference $D(A, B)$ between sequences A and B,

$$D(A, B) = \sum_{i=1}^{Np} d(a_p, b_p)$$

is searched throughout all scales.

It is important to evaluate multiple factors in comparing two shapes to achieve good matching results. For example, our dissimilarity measure evaluates four factors: amplitude, width, phase, and height, and it produces intuitively good matching results that fit with the human perception. However, since the local dissimilarity is obtained as a maximum of these four components, and what each factor evaluates is essentially different, it is better not to directly use $D(A, B)$ as the resultant dissimilarity of sequences A and B. Therefore, we employ the following strategy in determining the final difference of two sequences to be outputed.

(1) Find the best set of segment pairs $\mathbf{P}$ according to the local dissimilarity measure $d(a_m^{(k)}, b_l^{(h)})$, consisting of the four components: a, w, p, h.

(2) After finding best set of segment pairs, we further calculate the accumulated difference of height, $D_h(A, B)$ defined by

$$D_h(A, B) = \sum_{i=1}^{Np} d_h(a_p, b_p)$$

then output $D_h(A, B)$ as the dissimilarity between A and B.

This enable us to uniform the dimension of the dissimilarity with that of signal height, which has better linearity and understandability as the dissimilarity measure.

4. New Smoothing Kernel

According to Lindeberg[9], the Gaussian kernel and its derivatives are the unique scale-space kernels that guarantee non-creation of new local extrema with increasing scales. This property holds for continuous signals; namely, if $x(t)$ is continuous then convolution with a Gaussian kernel does not produce new local extrema with increasing scale. However, he also noted that if $x(t)$ is discrete, the translation from an arbitrary low scale to an arbitrary high scale is in general not a scale-space translation,

because the semi-group property of Gaussian function is not preserved after discretization. Besides, the central coefficient of sampled Gaussian kernel can be substantially large for small values of σ, preventing the detailed representation of original signals. He introduced the following new discrete kernel that solved the problems.

$$X(t,\sigma) = \sum_{n=-\infty}^{\infty} e^{-\sigma} I_n(\sigma) x(t-n)$$

where $e^{-\sigma} I_n(\sigma)$ is the new kernel and $I_n(\sigma)$ is the modified Bessel function of order n. Then we obtain the first- and second-order derivatives as follows.

$$X^{'}(t,\sigma) = \sum_{n=-\infty}^{\infty} -\frac{n}{\sigma} e^{-\sigma} I_n(\sigma) x(t-n)$$

$$X^{''}(t,\sigma) = \sum_{n=-\infty}^{\infty} \frac{1}{\sigma}(\frac{n^2}{\sigma}-1) e^{-\sigma} I_n(\sigma) x(t-n)$$

We have replaced the convolution formula in Section 2 by the above formula.

5. Experimental Results

We examined the usefulness of the proposed method on the cylinder-bell-funnel data set[10)11)], a simple synthetic data set which is well-known and frequently used in the temporal data mining community. Experiments were performed as follows. (1) Generate a data set containing a total of 384 sequences; 128 sequences for each of the three classes, cylinder, bell, and funnel. (2) Compute the dissimilarities for all pairs of sequences in the data sets using the proposed method. This produced a 384×384 symmetric dissimilarity matrix. (3) Remove one sequence, and predict its class according to the class label of the nearest sequence. The nearest sequence is selected according to the dissimilarity matrix. (4) Repeat step (3) for each of the 384 sequences, and evaluate the prediction error rate. Namely, we performed the leave-one-out validation with 1-Nearest Neighbor classification algorithm, using the dissimilarity matrix obtained by the proposed method as in[12)].

Before applying MSM, all of the input sequences were normalized in both horizontal and vertical directions by dividing by their standard deviation (because the length of sequences in cylinder-bell-funnel dataset were all the same, we simply normalized them in horizontal direction by dividing their length). The parameters in MSM were set as follows: starting scale = 0.1, scale interval = 0.5, number of scales = 100, cost weighting $\gamma = 0.15$.

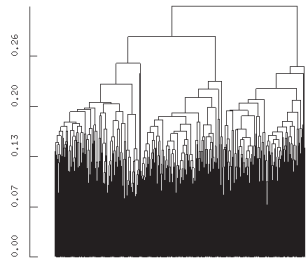

Fig. 3 A dendrogram generated by the proposed method for the CBF dataset.

The error rate was 0.054, which is quite better compared to the results summarized in[12)] (2nd best below the Euclidean distance, whose error rate = 0.003; we also reproduced the same result).

Next, we evaluated whether the dissimilarity matrix can be used to form meaningful clusters. We modified parts (3)-(4) of the above experimental procedures as follows. (3) remove one sequence, and using the 383×383 matrix, perform conventional average-linkage agglomerative clustering[13)] specifying the number of clusters to 3. (4) assign class label to each of the three clusters by the voting. (5) Perform 1-Nearest Neighbor classification for the removed sequence, and evaluate the classification accuracy. (6) remove another sequence and perform the same procedure. This is applied to all the 384 sequences.

The error rate was 0.042, similar to the previous experiment. We also performed the same experiments using the Euclidean distance, and its error rate was 0.216. This relatively high error rate of Euclidean distance implied that the dissimilarity matrix failed to form the clusters representing correct class distributions. Figure 3 provides a dendrogram of the entire dataset generated in combination with the MSM-based dissimilarity matrix and conventional average-linkage agglomerative clustering method (same as above procedure). It can be seen that the three-class structure of this data set was correctly represented. By setting an appropriate cutting point on the dendrogram, we can ob-

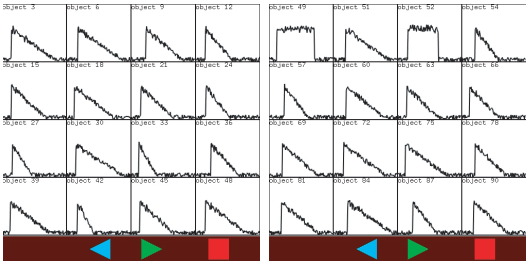

Fig. 4 Examples of sequences in cluster 1.

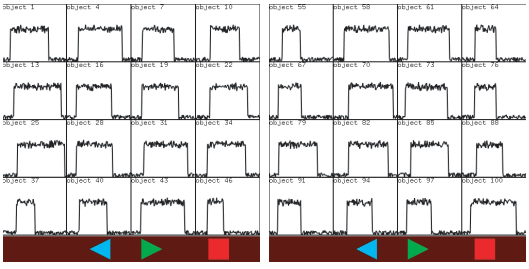

Fig. 5 Examples of sequences in cluster 2.

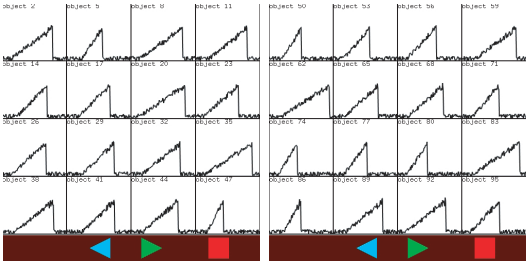

Fig. 6 Examples of sequences in cluster 3.

tain three clusters: cluster 1 (133 seqs; 128 funnel, 2 cylinder, 3 bell), 2 (125 seqs; 125 cylinder) and 3 (126 seqs; 126 bell). Figure 4-6 respectively show examples of sequences in these three clusters. Although several sequences belonged to cylinder or bell were missclustered into the 'funnel' cluster, the high accuracy ($> 98.7\%$) and highly reproductive accuracy (leave-one-out, $>95.8\%$) demonstrate the high performance of the proposed method in terms of accuracy.

6. Conclusions

In this paper, we have presented the modified multiscale matching as a new comparison method of time series. We have introduced a new representation of segment parameters directly induced from the base segments, which enables us to elude the problem of shrinkage occurring at high scales. Experiments on the CBF dataset demonstrated that the dissimilarity matrix produced by the proposed method, combined with conventional clustering techniques, lead to the successful clustering. It remains as a future work to validate the usefulness of the method on the real-world datasets.

Acknowledgments This work was supported in part by the Grant-in-Aid for Scientific Research on Priority Area (2)(#13131208) by the Ministry of Education, Culture, Science and Technology of Japan.

References

1) E. Keogh (2001): Mining and Indexing Time Series Data. Tutorial at IEEE ICDM-2001.
2) D. Sankoff and J. Kruskal (1999): Time Warps, String Edits, and Macromolecules. CLSI Publications.
3) S. Chu, E. J. Keogh, D. Hart, and M. J. Pazzani (2002): Iterative Deepening Dynamic Time Warping for Time Series. Proc. the Second SIAM Int'l Conf. on Data Mining.
4) K. P. Chan and A. W. Fu (1999): Efficient Time Series Matching by Wavelets. Proc IEEE ICDE-1999: 126–133.
5) S. Hirano and S. Tsumoto: Mining Similar Temporal Patterns in Long Time-series Data and Its Application to Medicine. Proc. IEEE ICDM-2002, Maebashi, 219–226.
6) F. Mokhtarian and A. K. Mackworth (1986): Scale-based Description and Recognition of planar Curves and Two Dimensional Shapes. IEEE Trans. PAMI, PAMI-8(1): 24-43.
7) N. Ueda and S. Suzuki (1990): A Matching Algorithm of Deformed Planar Curves Using Multiscale Convex/Concave Structures. IEICE Trans. Inf. and Syst., J73-D-II(7): 992–1000.
8) Lowe, D.G (1980): Organization of Smooth Image Curves at Multiple Scales. Int. J. Computer Vision, 3:119–130.
9) T. Lindeberg (1990): Scale-Space for Discrete Signals. IEEE Trans. PAMI, 12(3), 234–254.
10) N. Saito (1994): Local Feature Extraction and Its Application using a Library of Bases. Ph.D. Thesis, Yale University.
11) P. Geurts (2001): Pattern Extraction for Time-Series Classification. Proc. PAKDD-2001, 115-127.
12) E. Keogh and S. Kasetty (2003): On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. Data Mining And Knowledge Discovery 7:349-371.
13) B. S. Everitt, S. Landau, and M. Leese (2001): Cluster Analysis Fourth Edition. Arnold Publishers.