Multiscale Comparison of Temporal Patterns in Time-Series Medical Databases

Shoji Hirano and Shusaku Tsumoto

Department of Medical Informatics, Shimane Medical University, School of Medicine
89–1 Enya-cho, Izumo, Shimane 693–8501, Japan
hirano@ieee.org

Abstract. This paper presents a method for analyzing time-series data on laboratory examinations based on phase-constraint multiscale matching and rough clustering. Multiscale matching compares two subsequences throughout various scales of view. It has an advantage of preserving connectivity of subsequences even if the subsequences are represented at different scales. Rough clustering groups up objects according not to the topographic measures such as the center or deviance of objects in a cluster but to the relative similarity and indiscernibility of objects. We use multiscale matching to obtain similarity of sequences and rough clustering to cluster the sequences according to the obtained similarity. We slightly modified dissimilarity measure in multiscale matching so that it suppresses excessive shift of phase that may cause incorrect matching of the sequences. Experimental results on the hepatitis dataset show that the proposed method successfully clustered similar sequences into an independent cluster, and that correspondence of subsequences are also successfully captured.

Keywords: multiscale matching, rough clustering, rough sets, medical data mining, temporal knowledge discovery

1 Introduction

Since hospital information systems were first introduced in large hospitals in 1980's, huge amount of time-series laboratory examination data, for example blood and biochemical examination data, have been stored in the databases. Recently, analysis of such temporal examination databases has been attracting much interests because it might reveal underlying relationships between temporal course of examination and onset of diseases. Long-term laboratory examination databases might also enable us to validate a hypothesis about temporal course of chronic diseases that has not been evaluated yet on large samples. However, despite their importance, time-series medical databases have not widely been considered as the subject of analysis. This is primarily due to inhomogeneity of the data. Basically, the data were collected without considering further use in automated analysis. Therefore it involves the following problems. (1) Missing values: Examinations are not performed on every day when a patient comes to

T. Elomaa et al. (Eds.): PKDD, LNAI 2431, pp. 188–199, 2002.

the hospital. It depends on the needs for examination. (2) Irregular interval of data acquisition: A patient consults a doctor in different interval depending on his/her condition, hospital's vacancy, and other factors. The intervals can vary from a few days to several months. (3) Noise: The data can be distorted due to contingent change of patient's condition. These problems make it difficult to compare similarity of temporal patterns on different patients. Therefore, the data have been mainly used for visual comparison among small samples, where the scale-merits of large temporal databases have not been exploited.

In this paper, we present a hybrid approach to the analysis of such inhomogeneous time-series medical databases. The techniques employed here are phase-constraint multiscale structure matching [1] and rough-sets based clustering technique [2]. The first one, multiscale structure matching, is a method that effectively compares two objects by partially changing observation scales. We apply this method to the time-series data, and examine similarity of two sequences in both long-term and short-term points of view. It has an advantage that connectivity of segments is preserved in the matching results even when the partial segments are obtained from different scales. We slightly modified dissimilarity measure in multiscale matching so that it suppresses excessive shift of phase that causes incorrect matching results. The second technique, rough-sets based clustering, clusters sequences based on their indiscernibility defined in the context of rough set theory [3]. The method can produce interpretable clusters even under the condition that similarity of objects is defined only as a relative similarity. Our method attempts to cluster the temporal sequences according to their long- and short-term similarity by combining the two techniques. First, we apply multiscale stricture matching to all pairs of sequences and obtain similarity for each of them. Next, we apply rough-sets based clustering technique to cluster the sequences based on the obtained similarity. After then, common patterns in the clustered sequences can be visualized to understand relations to the diagnostic classes.

The remaining part of this paper is organized as follows. In Section 2 we introduce some related work. In Section 3 we describe the procedure of our method including explanation of each process such as preprocessing of data, multiscale structure matching and rough sets-based clustering. Then we show some experimental results in Section 4 and finally conclude the technical results.

2 Related Work

Data mining in time-series data has received much interests in both theoretical and applicational areas. A widely used approach in time-series data mining is to cluster sequences based on the similarity of their primary coefficients. Agrawal et al. [4] utilize discrete Fourier transformation (DFT) coefficients to evaluate similarity of sequences. Chan et al. [5] obtain the similarity based on the frequency components derived by the discrete wavelet transformation (DWT). Korn et al. [6] use singular value decomposition (SVD) to reduce complexity of sequences and compare the sequences according to the similarity of their eigen-

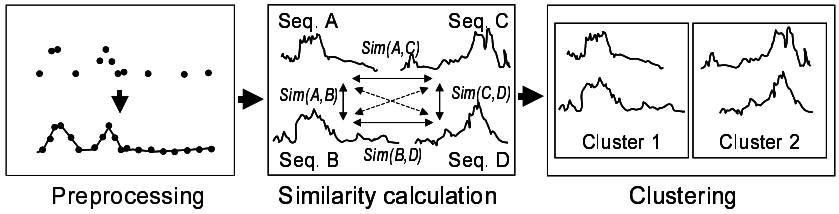

Fig. 1. Overview of the method

waves. Another approach includes comparison of sequences based on the similarity of forms of partial segments. Morinaka et al. [7] propose the L-index, which performs piecewise comparison of linearly approximated subsequences. Keogh et al. [8] propose a method called piecewise aggregate approximation (PAA), which performs fast comparison of subsequences by approximating each subsequence with simple box waves having constant length.

These methods can compare the sequences in various scales of view by choosing proper set of frequency components, or by simply changing size of the window that is used to translate a sequence into a set of simple waves or symbols. However, they are not designed to perform cross-scale comparison. In cross-scale comparison, connectivity of subsequences should be preserved across all levels of discrete scales. Such connectivity is not guaranteed in the existing methods because they do not trace hierarchical structure of partial segments. Therefore, similarity of subsequences obtained on different scales can not be directly merged into the resultant sequences. In other words, one can not capture similarity of sequences by partially changing scales of observation.

On the other hand, clustering has a rich history and a lot of methods have been proposed. They include, for example, k-means [9], fuzzy c-means [10], EM algorithm [11], CLIQUE [12], CURE [13] and BIRCH [14]. However, the similarity provided by multiscale matching is relative and not guaranteed to satisfy triangular inequality. Therefore, the methods based on the center, gravity or other types of topographic measures can not be applied to this task. Although classical agglomerative hierarchical clustering [15] can treat such relative similarity, in some case it has a problem that the clustering result depends on the order of handling objects.

3 Methods

3.1 Overview

Figure 1 shows an overview of the proposed method. First, we apply pre-processing to all the input sequences and obtain the interpolated sequences resampled in a regular interval. This procedure rearranges all data on the same time-scale and is required to compare long- and short-term difference using their length of trajectory. A simple linear interpolation of nearest neighbors is used to fill in a

missing value. Next, we apply multiscale structure matching to all possible combinations of two sequences and obtain their similarity as a matching score. We here restricted combinations of pairs so that they have the same attributes such as GPT-GPT, because our interest is not on the cross-attributes relationships. After obtaining similarity of the sequences, we cluster the sequences by using rough-set based clustering. Consequently, the similar sequences are clustered into the same clusters and their features are visualized.

3.2 Phase-Constraint Multiscale Structure Matching

Multiscale structure matching, proposed by Mokhtarian [17], is a method to describe and compare objects in various scales of view. Its matching criterion is similarity between partial contours. It seeks the best pair of partial contours throughout all scales, not only in the same scale. This enables matching of object not only from local similarity but also from global similarity. The method required much computation time because it should continuously change the scale, however, Ueda et al. [1] solved this problem by introducing a segment-based matching method which enabled the use of discrete scales. We use Ueda's method to perform matching of time sequences between patients. We associate a convex/concave structure in the time-sequence as a convex/concave structure of partial contour. Such a structure can be generated by increase/decrease of examination values. Then we can compare the sequences from different terms of observation.

Now let $x(t)$ denote a time sequence where t denotes time of examination. The sequence at scale σ, $X(t,\sigma)$, can be represented as a convolution of $x(t)$ and a Gauss function with scale factor σ, $g(t,\sigma)$, as follows:

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

Figure 2 shows an example of sequences in various scales. From Figure 2 and the function above, it is obvious that the 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 derivative 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

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

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 partial contour whose ends

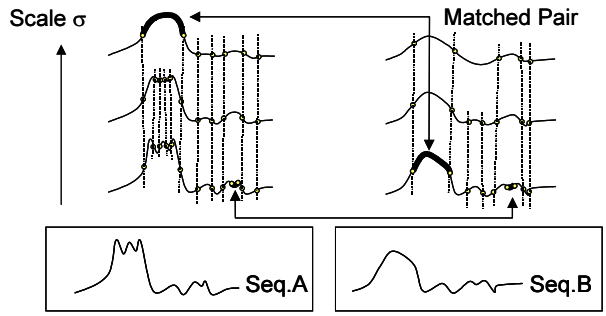

Fig. 2. Multiscale matching

correspond to the adjacent inflection points. Let $\mathbf{A}^{(k)}$ be a set of N segments that represents the sequence at scale $\sigma^{(k)}$ as

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

Then, difference between segments $a_i^{(k)}$ and $b_i^{(h)}$, $d(a_i^{(k)}, b_j^{(h)})$ is defined as follows:

$$d(a_i^{(k)}, b_j^{(h)}) = \frac{\mid \theta_{a_i}^{(k)} - \theta_{b_j}^{(h)} \mid}{\theta_{a_i}^{(k)} + \theta_{b_j}^{(h)}} \left| \frac{l_{a_i}^{(k)}}{L_A^{(k)}} - \frac{l_{b_j}^{(h)}}{L_B^{(h)}} \right|,$$

where $\theta_{a_i}^{(k)}$ and $\theta_{b_j}^{(h)}$ denote rotation angles of tangent vectors along the contours, $l_{a_i}^{(k)}$ and $l_{b_j}^{(h)}$ denote length of the contours, $L_A^{(k)}$ and $L_B^{(h)}$ denote total segment length of the sequences A and B at scales $\sigma^{(k)}$ and $\sigma^{(h)}$. According to the above definition, large differences can be assigned when difference of rotation angle or relative length is large. Continuous $2n-1$ segments can be integrated into one segment at higher scale. Difference between the replaced segments and another segment can be defined analogously, with additive replacement cost that suppresses excessive replacement.

The above similarity measure can absorb shift of time and difference of sampling duration. However, we should suppress excessive back-shift of sequences in order to correctly distinguish the early-phase events from late-phase events. Therefore, we extend the definition of similarity as follows.

$$d(a_i^{(k)}, b_j^{(h)}) = \frac{1}{3} \left(\left| \frac{d_{a_i}^{(k)}}{D_A^{(k)}} - \frac{d_{b_j}^{(h)}}{D_B^{(h)}} \right| + \frac{\mid \theta_{a_i}^{(k)} - \theta_{b_j}^{(h)} \mid}{\theta_{a_i}^{(k)} + \theta_{b_j}^{(h)}} + \left| \frac{l_{a_i}^{(k)}}{L_A^{(k)}} - \frac{l_{b_j}^{(h)}}{L_B^{(h)}} \right| \right),$$

where $d_{a_i}^{(k)}$ and $d_{b_j}^{(h)}$ denote dates from first examinations, $D_A^{(k)}$ and $D_B^{(h)}$ denote durations of examinations. By this extension, we can simultaneously evaluate the

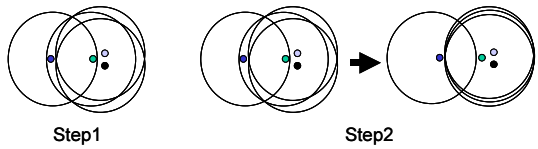

Fig. 3. Rough clustering

following three similarities: (1)dates of events (2) velocity of increase/decrease (3) duration of each event.

The remaining procedure of multiscale structure matching is to find the best pair of segments that minimizes the total difference. Figure 2 illustrates the process. For example, in the upper part of Figure 2, five contiguous segments at the lowest scale of Sequence A are integrated into one segment at the highest scale, and this segment is well matched to one segment in Sequence B at the lowest scale. While, another pair of segments is matched at the lowest scale. In this way, matching is performed throughout all scales. The matching process can be fasten by implementing dynamic programming scheme. For more details, see ref [1]. After matching process is completed, we calculate the remaining difference and use it as a measure of similarity between sequences.

3.3 Rough-Sets Based Clustering

Generally, if similarity of objects is represented only as a relative similarity, it is not an easy task to construct interpretable clusters because some of important measures such as inter- and intra-cluster variances are hard to be defined. The rough-set based clustering method is a clustering method that clusters objects according to the indiscernibility of objects. It represents denseness of objects according to the *indiscernibility degree*, and produces interpretable clusters even for the objects mentioned above. Since similarity of sequences obtained through multiscale structure matching is relative, we use this clustering method to classify the sequences.

The clustering method lies its basis on the *indiscernibility* of objects, which forms basic property of knowledge in rough sets. Let us first introduce some fundamental definitions of rough sets related to our work. Let $U \neq \phi$ be a universe of discourse and X be a subset of U. An equivalence relation, R, classifies U into a set of subsets $U/R = \{X_1, X_2, ...X_m\}$ in which following conditions are satisfied:

$$\begin{array}{ll} (1) X_i \subseteq U, X_i \neq \phi & \text{for any } i, \\ (2) X_i \cap X_j = \phi & \text{for any } i, j, \\ (3) \cup_{i=1,2,...n} X_i = U. & \end{array}$$

Any subset X_i, called a category, represents an equivalence class of R. A category in R containing an object $x \in U$ is denoted by $[x]_R$. For a family of equivalence relations $\mathbf{P} \subseteq \mathbf{R}$, an indiscernibility relation over $\mathbf{P}$ is denoted by $IND(\mathbf{P})$ and

defined as follows

$$IND(\mathbf{P}) = \{(x_i, x_j) \in U^2 \mid \forall Q \in \mathbf{P}, [x_i]_Q = [x_j]_Q\}.$$

The clustering method consists of two steps: (1)assignment of initial equivalence relations and (2)iterative refinement of initial equivalence relations. Figure 3 illustrates each step. In the first step, we assign an initial equivalence relation to every object. An initial equivalence relation classifies the objects into two sets: one is a set of objects similar to the corresponding objects and another is a set of dissimilar objects. Let $U = \{x_1, x_2, ..., x_n\}$ be the entire set of n objects. An initial equivalence relation R_i for object x_i is defined as

$$R_i = \{\{P_i\}, \{U - P_i\}\},$$

$$P_i = \{x_j |\ s(x_i, x_j) \geq S_i\}, \quad \forall x_j \in U.$$

where P_i denotes a set of objects similar to x_i. Namely, P_i is a set of objects whose similarity to x_i, s, is larger than a threshold value S_i. Here, s corresponds to the inverse of the output of multiscale structure matching, and S_i is determined automatically at a place where s largely decreases. A set of indiscernible objects obtained using all sets of equivalence relations corresponds to a cluster. In other words, a cluster corresponds to a category X_i of $U/IND(\mathbf{R})$.

In the second step, we refine the initial equivalence relations according to their global relationships. First, we define an indiscernibility degree, γ, which represents how many equivalence relations commonly regards two objects as indiscernible objects, as follows:

$$\gamma(x_i, x_j) = \frac{1}{|U|} \sum_{k=1}^{|U|} \delta_k(x_i, x_j),$$

$$\delta_k(x_i, x_j) = \begin{cases} 1, \text{ if } [x_k]_{R_k} \cap ([x_i]_{R_k} \cap [x_j]_{R_k}) \neq \phi \\ 0, \text{ otherwise.} \end{cases}$$

Objects with high indiscernibility degree can be interpreted as similar objects. Therefore, they should be classified into the same cluster. Thus we modify an equivalence relation if it has ability to discern objects with high γ as follows:

$$R'_i = \{\{P'_i\}, \{U - P'_i\}\},$$

$$P'_i = \{x_j | \gamma(x_i, x_j) \geq T_h\}, \quad \forall x_j \in U.$$

This prevents generation of small clusters formed due to the too fine classification knowledge. T_h is a threshold value that determines indiscernibility of objects. Therefore, we associate T_h with roughness of knowledge and perform iterative refinement of equivalence relations by constantly decreasing Th. Consequently, coarsely classified set of sequences are obtained as $U/IND(\mathbf{R}')$.

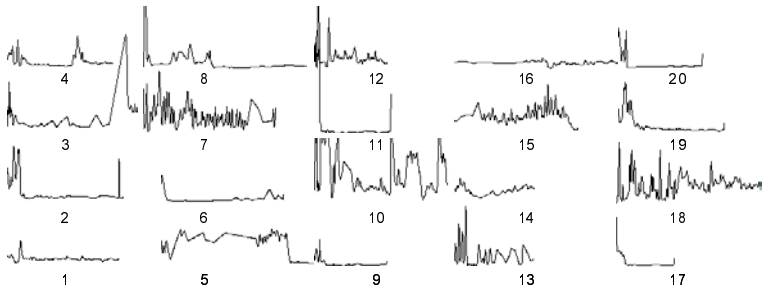

Fig. 4. Test patterns

Table 1. Similarity of the sequences

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1.00	0.70	0.68	0.78	0.00	0.63	0.48	0.71	0.72	0.61	0.73	0.66	0.64	0.72	0.50	0.00	0.53	0.00	0.74	0.45
2		1.00	0.61	0.73	0.00	0.68	0.22	0.46	0.68	0.67	0.72	0.73	0.72	0.68	0.54	0.00	0.68	0.00	0.77	0.41
3			1.00	0.75	0.45	0.51	0.68	0.47	0.71	0.70	0.69	0.73	0.71	0.81	0.68	0.00	0.62	0.00	0.72	0.55
4				1.00	0.00	0.60	0.52	0.47	0.75	0.71	0.64	0.79	0.75	0.82	0.47	0.00	0.60	0.00	0.75	0.48
5					1.00	0.23	0.62	0.49	0.33	0.53	0.44	0.45	0.50	0.44	0.56	0.01	0.00	0.26	0.53	0.30
6						1.00	0.00	0.00	0.59	0.00	0.58	0.39	0.61	0.65	0.00	0.00	0.47	0.00	0.47	0.48
7							1.00	0.49	0.54	0.80	0.57	0.73	0.73	0.59	0.76	0.00	0.00	0.44	0.62	0.39
8								1.00	0.53	0.47	0.57	0.56	0.51	0.49	0.54	0.00	0.00	0.00	0.66	0.51
9									1.00	0.00	0.68	0.00	0.00	0.00	0.00	0.00	0.82	0.00	0.00	0.00
10										1.00	0.59	0.83	0.76	0.75	0.81	0.00	0.47	0.11	0.59	0.37
11											1.00	0.76	0.54	0.68	0.00	0.00	0.74	0.00	0.76	0.00
12												1.00	0.81	0.78	0.67	0.00	0.70	0.00	0.63	0.40
13													1.00	0.75	0.00	0.00	0.64	0.00	0.67	0.35
14														1.00	0.00	0.00	0.66	0.00	0.71	0.00
15															1.00	0.00	0.43	0.20	0.55	0.39
16																1.00	0.00	0.00	0.43	0.19
17																	1.00	0.00	0.00	0.00
18																		1.00	0.39	0.03
19																			1.00	0.00
20																				1.00

4 Experimental Results

We applied the proposed method to time-series GPT sequences in the hepatitis data set [18]. The dataset contained long time-series data on laboratory examinations, which were collected on a university hospital in Japan. The subjects were 771 patients of hepatitis B and C who took examinations between 1982 and 2001. Due to incompleteness in data acquisition, time-series GPT sequences were available only for 195 of 771 patients.

First, in order to evaluate applicability of multiscale matching to time-series data analysis, we applied the proposed method to a small subset of sequences which was constructed by randomly selecting 20 sequences from the data set. Figure 4 shows all the pre-processed sequences. Each sequence originally has different sampling intervals from one day to one year. From preliminary analysis we found that the most frequently appeared interval was one week; this means that most of the patients took examinations on a fixed day of a week. According to this observation, we determined resampling interval to seven days.

Table 1 shows normalized similarity of the sequences derived by multiscale matching. Since consistency of self-similarity ($s(A, B) = s(B, A)$) holds, the

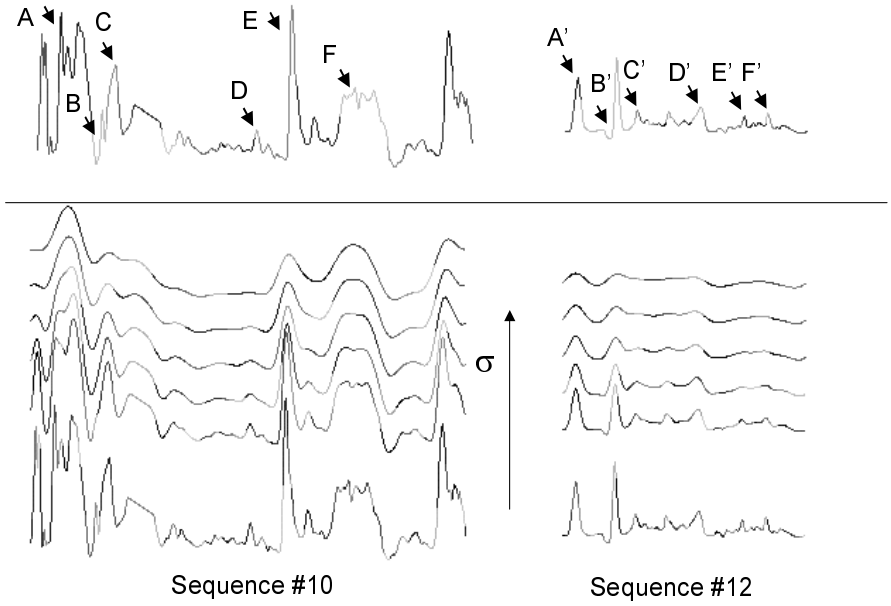

Fig. 5. Matching result of sequences #10 and #12

lower-left half of the matrix is omitted. We can observe that higher similarity was successfully assigned to intuitively similar pairs of sequences.

Based on this similarity, the rough clustering produced nine clusters: $U/IND(\mathbf{R}) = \{\{1,2,9,11,17,19\}, \{4,3,8\}, \{7,14,15\}, \{10,12,13\}, \{5\}, \{6\}, \{16\}, \{18\}, \{20\}\}$. A parameter Th for rough clustering was set to $Th = 0.6$. Refinement was performed up to five times with constantly decreasing Th toward $Th = 0.4$. It can be seen that similar sequences were clustered into the same cluster. Some sequences, for example #16, were clustered into independent clusters due to remarkably small similarity to other sequences. This is because multiscale matching could not find good pairs of subsequences.

Figure 5 shows the result of multiscale matching on sequences #10 and #12, that have high similarity. We changed σ from 1.0 to 13.5, with intervals of 2.5. At the bottom of the figure there are original two sequences at $\sigma = 1.0$. The next five sequences represent sequences at scales $\sigma = 3.5$, 6.0, 8.5, 11.0, and 13.5, respectively. Each of the colored line corresponds to a segment. The matching result is shown at the top of the figure. Here the lines with same color represent the matched segments, for example, segment A matches segment A' and segment B matches segment B'. We can clearly observe that increase/decrease patterns of sequences are successfully captured; large increase (A and A'), small decrease with instant increase (B and B'), small increase (C and C') and so on. Segments $D - F$ and $D' - F'$ have similar patterns and the feature was

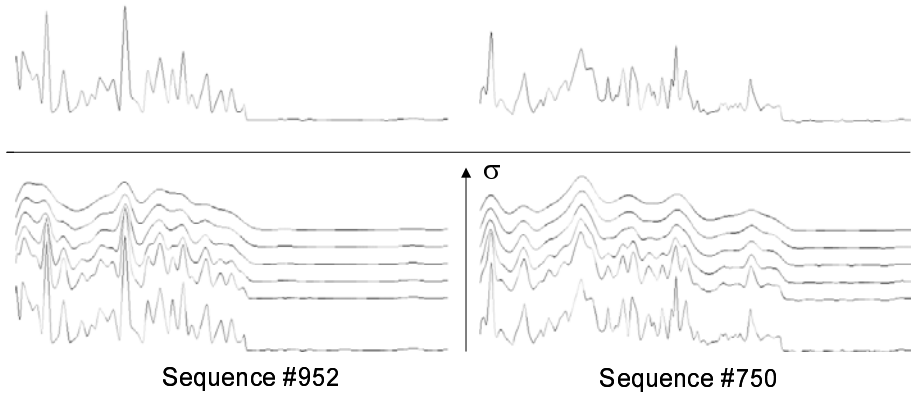

Fig. 6. Matching result of sequences #952 and #750

also correctly captured. It can also be seen that the well-matched segments were obtained in the sequences with large time difference.

Next, we applied the proposed method to the full data set containing 195 GPT sequences. For this data set, rough clustering produced 14 clusters: $U/IND(\mathbf{R}) = \{\{2, 19, 36, 37, 49, \ldots, 953, 955$ (total 165 sequences)$\}, \{16, 35, 111\}, \{86, 104, 142, 171, 215, 273, 509, 523, 610, 663\}, \{149\}, \{703\}, \{706\}, \{737\}, \{740\}, \{743, 801, 894, 897, 942\}, \{750, 952\}, \{771\}, \{533, 594\}, \{689, 731\}\}$, where a sequence number corresponds to a masked ID of the patient. The first cluster seems to be uninteresting because it contains too many sequences. This cluster was generated as a result of inproper assignment of Th, which caused excessive refinement of clusters. However instead, we could find very interesting patterns in other clusters. For example, the 10th cluster contained sequences 750 and 952, which had very similar patterns as shown in Figure 6. In both sequences, increase and decrease of GPT values were repeatedly observed in the early half period of data acquisition, and they became flat in the late period. A physician evaluated that this might be an interesting pattern that represents degree of damage of the liver.

5 Conclusions

In this paper, we have presented an analysis method of time-series medical databases based on the hybridization of phase-constraint multiscale structure matching and rough clustering. The method first obtained similarity of sequences by multiscale comparison of sequences in which connectivity of subsequences were preserved even if they were represented at different scales. Then rough clustering grouped up the sequences according to their relative similarity. This hybridization enabled us not only to cluster time-series sequence from both long- and short-term viewpoints but also to visualize correspondence of subsequences.

In the experiments on the hepatitis data set, we showed that the sequences were successfully clustered into intuitively correct clusters, and that some interesting patterns were discovered by visualizing the clustered sequences. It remains as a future work to evaluate usefulness of the method in other databases.

Acknowledgment

This work was supported in part by the Grant-in-Aid for Scientific Research on Priority Area (B)(No.759) "Implementation of Active Mining in the Era of Information Flood" by the Ministry of Education, Culture, Science and Technology of Japan.

References

1. N. Ueda and S. Suzuki (1990): A Matching Algorithm of Deformed Planar Curves Using Multiscale Convex/Concave Structures. IEICE Transactions on Information and Systems, J73-D-II(7): 992–1000. 189, 191, 193
2. S. Hirano and S. Tsumoto (2001): Indiscernibility Degrees of Objects for Evaluating Simplicity of Knowledge in the Clustering Procedure. Proceedings of the 2001 IEEE International Conference on Data Mining. 211–217. 189
3. Z. Pawlak (1991): Rough Sets, Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht. 189
4. R Agrawal, C. Faloutsos, and A. N. Swami (1993): Efficient Similarity Search in Sequence Databases. Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms: 69–84. 189
5. K. P. Chan and A. W. Fu (1999): Efficient Time Series Matching by Wavelets. Proceedings of the 15th IEEE International Conference on Data Engineering: 126–133. 189
6. F. Korn, H. V. Jagadish, and C. Faloutsos (1997): Efficiently Supporting Ad Hoc Queries in Large Datasets of Time Sequences. Proceedings of ACM SIGMOD International Conference on Management of Data: 289–300. 189
7. Y. Morinaka, M. Yoshikawa, T. Amagasa and S. Uemura (2001): The L-index: An Indexing Structure for Efficient Subsequence Matching in Time Sequence Databases. Proceedings of International Workshop on Mining Spatial and Temporal Data, PAKDD-2001: 51-60. 190
8. E. J. Keogh, K. Chakrabarti, M. J. Pazzani, and S. Mehrotra (2001): "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases" Knowledge and Information Systems 3(3): 263-286. 190
9. S. Z. Selim and M. A. Ismail (1984): K-means-type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality. IEEE Transactions on Pattern Analysis and Machine Intelligence, 6(1): 81–87. 190
10. J. C. Bezdek (1981): Pattern Recognition with Fuzzy Objective Function Algorithm. Plenum Press, New York. 190
11. A. P. Dempster, N. M. Laird, and D. B. Rubin (1977): Maximum likelihood from incomplete data via the EM algorithm. J. of Royal Statistical Society Series B, 39: 1–38. 190

12. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan (1998): Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. Proceedings of ACM SIGMOD International Conference on Management of Data: 94–105. 190
13. S. Guha, R. Rastogi, and K. Shim(1998): CURE: An Efficient Clustering Algorithm for Large Databases. Proceedings of ACM SIGMOD International Conference on Management of Data: 73–84. 190
14. T. Zhang, R. Ramakrishnan, and M. Livny (1996): BIRCH: An Efficient Data Clustering Method for Very Large Databases. Proceedings of ACM SIGMOD International Conference on Management of Data: 103–114. 190
15. M. R. Anderberg (1973): Cluster Analysis for Applications. Academic Press, New York. 190
16. R. H. Shumway and D. S. Stoffer (2000): Time Series Analysis and Its Applications. Springer-Verlag, New York.
17. F. Mokhtarian and A. K. Mackworth (1986): Scale-based Description and Recognition of planar Curves and Two Dimensional Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(1): 24-43. 191
18. URL: http://lisp.vse.cz/challenge/ecmlpkdd2002/ 195