Granularity in Temporal Data Mining

Shoji Hirano and Shusaku Tsumoto
Department of MedicalInformatics, Shimane University
89-1 Enya-cho, Izumo 693-8501 Japan
{tsumoto,hirano}@med.shimane-u.ac.jp

Abstract

This paper focuses on clustering of trajectories of temporal sequences of two laboratory examinations. First, we map a set of time series containing different types of laboratory tests into directed trajectories representing temporal change in patients' status. Then the trajectories for individual patients are compared in multiscale and grouped into similar cases by using clustering methods.

Keywords: Trajectory Analysis, Temporal Data Mining, .

1 Introduction

Hosptial information system (HIS) collects all the data from all the branches of departments in a hospital, including laboratory tests,physiological tests, electronic patient records. Thus, HIS can be viewed as a large heterogenous database, which stores chronological changes in patients' status. Recent advances not only in informaiton technology, but also other developments in devices enable us to collect huge amount of temporal data automatically, one of whose advantage is that we are able not only to analyze the data within one patient, but also the data in a cross-sectoral manner. It may reveal a underlying mechanism in temporal evolution of (chronic) diseases with some degree of evidence, which can be used to predict or estimate a new case in the future. Especially, finding temporally covariant variables is very important for clinical practice because we are able to obtain the measurements of some examinations very easily, while it takes a long time for us to measure other ones. Also, unexpected covariant patterns give us new knowledge for temporal evolution of chronic diseases. However, despite of its importance, large-scale analysis of time-series medical databases has rarely been reported due to the following problems: (1) sampling intervals and lengths of data can be both irregular, as they depend on the condition of each patient. (2) a time series can include various types of events such as acute changes and chronic changes. When comparing the time series, one is required to appropriately determine the correspondence of data points to be compared taking into account the above issues. Additionally, the dimensionality of data can be usually high due to the variety of medical examinations. These fearures prevent us from using conventional time series analysis methods.

This paper presents a novel cluster analysis method for multivariate time-series data on medical laboratory tests. Our method represents time series of test results as trajectories in multidimensional space, and compares their structural similarity by using the multiscale comparison technique [1]. It enables us to find the part-to-part correspondences between two trajectories, taking into account the relationships between different tests. The resultant dissimilarity can be further used as input for clustering algorithms for finding the groups of similar cases. In the experiments we demonstrate the usefulness of our approach through the grouping tasks of artificially generated digit stroke trajectories and medical test trajectories on chronic hepatitis patients.

L. Magdalena, M. Ojeda-Aciego, J.L. Verdegay (eds): Proceedings of IPMU'08, pp. 1779–1785
Torremolinos (Málaga), June 22–27, 2008

The remainder of this paper is organized as follows. In Section 2 we describe the methodoology, including preprocessing of the data. In Section 3 we show experimental results on a synthetic data (digit strokes) and chronic hepatitis data (albumin-platelet trajectories and cholinesterase-platelet trajectories). Finally, Section 4 is a conclusion of this paper.

2 Methods

2.1 Overview

Figure 1 1 shows an overview of the whole process of clustering of trajectories. First, we apply preprocessing of a raw temporal sequence for each variable (Subsection 2.2). Secondly, a trajectory of laboratory tests is calculated for each patient, segmentation technique is applied to each sequence for generation of a segmentation hiearchy (Subsection 2.3). Third, we trace segemented sequences and search for matching between two sequences in a hiearchical way (Subsection 2.4). Then, dissimilarities are calculated for matched sequences (Subsection 2.5 and 2.6). Finally, we apply clustering to the dissimilarities obtained (Subsection 2.7).

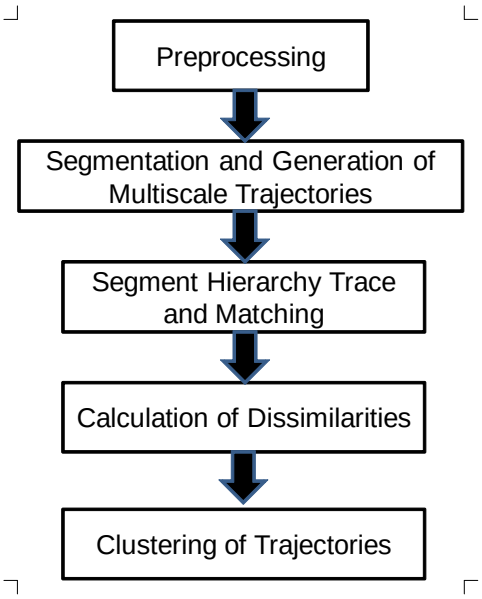

Figure 1: Overview of Trajectory Clustering

2.2 Preprocessing

Time-series examination data is often represented as a tuple of examination date and results. Interval of examinations is usually irregular, as it depends on the condition of a patient. However, in the process of multiscale matching, it is neccessary to represent time-series as a set of data points with a constant interval in order to represent the time span by the number of data points. Therefore, we employed linear interpolation and constructed new equi-interval data.

2.3 Multiscale Description of Trajectories by the Modified Bessel Function

Let us consider examination data for one person, consisting of I different time-series examinations. Let us denote the time series of i-th examination by $ex_i(t)$, where $i \in I$. Then the trajectory of examination results, $c(t)$ is denoted by

$$c(t) = \{ex_1(t), ex_2(t), \ldots, ex_I(t)\}$$

Next, let us denote an observation scale by σ and denote a Gaussian function with scale parameter σ^2 by $g(t, \sigma)$. Then the time-series of the i-th examination at scale σ, $EX_i(t, \sigma)$ is derived by convoluting $ex_i(t)$ with $g(t, \sigma)$ as follows.

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

Applying the above convolution to all examinations, we obtain the trajectory of examination results at scale σ, $C(t, \sigma)$, as

$$\begin{aligned} &C(t,\sigma) \\ &= \{EX_1(t,\sigma), EX_2(t,\sigma), \ldots, EX_I(t,\sigma)\} \end{aligned}$$

By changing the scale factor σ, we can represent the trajectory of examination results at various observation scales. Figure 2 illustrates an example of multiscale representation of trajectories where $I = 2$. Increase of σ induces the decrease of convolution weights for neighbors. Therefore, more flat trajectories with less inflection points will be observed at higher scales.

Curvature of the trajectory at time point t is defined by, for $I = 2$,

$$K(t,\sigma) = \frac{EX_1' EX_2'' + EX_1'' EX_2'}{(EX_1'^2 + EX_2'^2)^{3/2}}$$

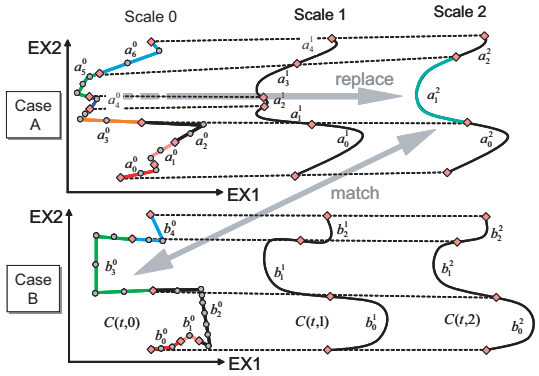

Figure 2: Multiscale representation and matching scheme.

where EX_i' and EX_i'' denotes the first- and second-order derivatives of $EX_i(t,\sigma)$ respectively. The m-th order derivative of $EX_i(t,\sigma)$, $EX_i^{(m)}(t,\sigma)$, is defined by

$$EX_i^{(m)}(t,\sigma) = \frac{\partial^m EX_i(t,\sigma)}{\partial t^m} = ex_i(t) \otimes g^{(m)}(t,\sigma)$$

It should be noted that many of the real-world time-series data, including medical data, can be discrete in time domain. Thus, a sampled Gaussian kernel is generally used for calculation of $EX_i(t,\sigma)$, changing an integral to summation. However, Lindeberg [2] pointed out that, a sampled Gaussian may lose some of the properties that a continuous Gaussian has, for example, non-creation of local extrema with the increase of scale. Additionally, in a sampled Gaussian kernel, the center value can be relatively large and imbalanced when the scale is very small. Ref. [2] suggests the use of kernel based on the modified Bessel function, as it is derived by incorporating the discrete property. Since this influences the description ability about detailed structure of trajectories, we employed the Lindeberg's kernel and derive $EX_i(t,\sigma)$ as follows.

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

where $I_n(\sigma)$ denotes the modified Bessel function of order n. The first- and second-order derivatives of $EX_i(t,\sigma)$ are obtained as follows.

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

$$EX_i''(t,\sigma) = \sum_{n=-\infty}^{\infty} \frac{1}{\sigma}\left(\frac{n^2}{\sigma} - 1\right) \times e^{-\sigma} I_n(\sigma) ex_i(t-n)$$

2.4 Segment Hierarchy Trace and Matching

For each trajectory represented by multiscale description, we find the places of inflection points according to the sign of curvature. Then we divide each trajectory into a set of convex/concave segments, where both ends of a segment correspond to adjacent inflection points. Let A be a trajectory at scale k composed of $M^{(k)}$ segments. Then A is represented by $\mathbf{A}^{(k)} = \{a_i^{(k)} \mid i = 1, 2, \cdots, M^{(k)}\}$, where $a_i^{(k)}$ denotes i-th segment at scale k. Similarly, another trajectory B at scale h is represented by $\mathbf{B}^{(h)} = \{b_j^{(h)} \mid j = 1, 2, \cdots, N^{(h)}\}$.

Next, we chase the cross-scale correspondence of inflection points from top scales to bottom scale. It defines the hierarchy of segments and enables us to guarantee the connectivity of segments represented at different scales. Details of the algorithm for checking segment hierarchy is available on ref. [1]. In order to apply the algorithm for closed curve to open trajectory, we modified it to allow replacement of odd number of segments at sequence ends, since cyclic property of a set of inflection points can be lost.

The main procedure of multiscale matching is to search the best set of segment pairs that satisfies both of the following conditions:

1. Complete Match: By concatenating all segments, the original trajectory must be completely formed without any gaps or overlaps.
2. Minimal Difference: The sum of segment dissimilarities over all segment pairs should be minimized.

The search is performed throughout all scales. For example, in Figure 2, three contiguous segments $a_3^{(0)} - a_5^{(0)}$ at the lowest scale of case A can

be integrated into one segment $a_1^{(2)}$ at upper scale 2, and the replaced segment well matches to one segment $b_3^{(0)}$ of case B at the lowest scale. Thus the set of the three segments $a_3^{(0)} - a_5^{(0)}$ and one segment $b_3^{(0)}$ will be considered as a candidate for corresponding segments. On the other hand, segments such as $a_6^{(0)}$ and $b_4^{(0)}$ are similar even at the bottom scale without any replacement. Therefore they will be also a candidate for corresponding segments. In this way, if segments exhibit short-term similarity, they are matched at a lower scale, and if they present long-term similarity, they are matched at a higher scale.

2.5 Local Segment Difference

In order to evaluate the structural (dis-)similarity of segments, we first describe the structural feature of a segment by using shape parameters defined below.

1. Gradient at starting point: $g(a_m^{(k)})$
2. Rotation angle: $\theta(a_m^{(k)})$
3. Velocity: $v(a_m^{(k)})$

Figure 3 illustrates these parameters. Gradient represents the direction of the trajectory at the beginning of the segment. Rotation angle represents the amount of change of direction along the segment. Velocity represents the speed of change in the segment, which is calculated by dividing segment length by the number of points in the segment.

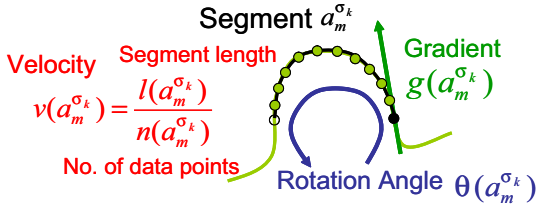

Figure 3: Segment Parameters.

Next, we define the local dissimilarity of two segments, $a_m^{(k)}$ and $b_n^{(h)}$, as shown in Fig. 4, where $cost()$ denotes a cost function used for suppressing excessive replacement of segments, and γ is the weight of costs. We define the cost function using local segment dissimilarity as follows. For a segment $a_m^{(k)}$ that replaces p segments $a_r^{(0)} - a_{r+p-1}^{(0)}$ at the bottom scale,

$$cost(a_m^{(k)}) = \sum_{q=r}^{r+p-1} d(a_q^{(0)}, a_{q+1}^{(0)})$$

2.6 Sequence Dissimilarity

After determining the best set of segment pairs, we newly calculate value-based dissimilarity for each pair of matched segments. The local segment dissimilarity defined in the previous section reflects the structural difference of segments, but does not reflect the difference of original sequence values; therefore, we calculate the value-based dissimilarity that can be further used as a metric for proximity in clustering.

Suppose we obtained L pairs of matched segments after multiscale matching of trajectories A and B. The value-based dissimilarity between A and B, $D_{val}(A, B)$, is defined as follows.

$$D_{val}(A, B) = \sum_{l=1}^{L} d_{val}(\alpha_l, \beta_l)$$

where α_l denotes a set of contiguous segments of A at the lowest scale that constitutes the l-th matched segment pair ($l \in L$), and β_l denotes that of B. For example, suppose that segments $a_3^{(0)} \sim a_5^{(0)}$ of A and segment $b_3^{(0)}$ of B in Figure 2 constitute the l-th matched pair. Then, $\alpha_l = a_3^{(0)} \sim a_5^{(0)}$ and $\beta_l = b_3^{(0)}$, respectively. $d_{val}(\alpha_l, \beta_l)$ is the difference between α_l and β_l in terms of data values at the peak and both ends of the segments. For the i-th examination ($i \in I$), $d_{val_i}(\alpha_l, \beta_l)$ is defined as

$$\begin{aligned} d_{val_i}(\alpha_l, \beta_l) &= peak_i(\alpha_l) - peak_i(\beta_l) \\ &\quad + \frac{1}{2}\{left_i(\alpha_l) - left_i(\beta_l)\} \\ &\quad + \frac{1}{2}\{right_i(\alpha_l) - right_i(\beta_l)\} \end{aligned}$$

where $peak_i(\alpha_l)$, $left_i(\alpha_l)$, and $right_i(\alpha_l)$ denote data values of the i-th examination at the peak, left end and right end of segment α_l, respectively. If α_l or β_l is composed of plural segments, the centroid of the peak points of those segments

$$d(a_m^{(k)}, b_n^{(h)}) = \sqrt{\left(g(a_m^{(k)}) - g(b_n^{(h)})\right)^2 + \left(\theta(a_m^{(k)}) - \theta(b_n^{(h)})\right)^2} + \left|v(a_m^{(k)}) - v(b_n^{(h)})\right| + \gamma\left\{cost(a_m^{(k)}) + cost(b_n^{(h)})\right\}$$

Figure 4: Formula for Local Dissimilarity of Two Segments

is used as the peak of α_l. Finally, d_{val_i} is integrated over all examinations as follows.

$$d_{val}(\alpha_l, \beta_l) = \frac{1}{I}\sqrt{\sum_i d_{val_i}(\alpha_l, \beta_l)}$$

2.7 Clustering

For clustering, we employ two methods: agglomerative hierarchical clustering (AHC) [3] and rough set-based clustering (RC) [4]. The sequence comparison part performs pairwise comparison for all possible pairs of time series, and then produces a dissimilarity matrix. The clustering part performs grouping of trajectories according to the given dissimilarity matrix.

3 Experimental Results

We applied our method to the chronic hepatitis dataset which was a common dataset in ECML/PKDD discovery challenge 2002-2004 [5]. The dataset contained time series laboratory examinations data collected from 771 patients of chronic hepatitis B and C. In this work, we focused on analyzing the temporal relationships between platelet count (PLT), albumin (ALB) and cholinesterase (CHE), that were generally used to examine the status of liver function. Our goals were set to: (1) find groups of trajectories that exhibit interesting patterns, and (2) analyze the relationships between these patterns and the stage of liver fibrosis.

We selected a total of 488 cases which had valid examination results for all of PLT, ALB, CHE and liver biopsy. Constitution of the subjects classified by virus types and administration of interferon (IFN) was as follows. Type B: 193 cases, Type C with IFN: 296 cases, Type C without IFN: 99 cases. In the following sections, we mainly describe the results about Type C without IFN cases, which contained the natural courses of Type C viral hepatitis.

Experiments were conducted as follows. This procedure was applied separately for CHE-PLT trajectories.

1. Select a pair of cases (patients) and calculate the dissimilarity by using the proposed method. Apply this procedure for all pairs of cases, and construct a dissimilarity matrix.

2. Create a dendrogram by using conventional hierarchical clustering [3] and the dissimilarity matrix. Then perform cluster analysis.

Parameters for multiscale matching were empirically determined as follows: starting scale = 0.5, scale interval = 0,5, number of scales = 100, weight for segment replacement cost = 1.0. We used group average as a linkage criterion for hierarchical clustering. The experiments were performed on a small PC cluster consisted of 8 DELL PowerEdge 1750 (Intel Xeon 2.4GHz 2way) workstations. It took about three minutes to make the dissimilarity matrix for all cases.

3.1 Results on CHE-PLT trajectories

Figure 5 shows the dendrogram generated from CHE-PLT trajectories of 99 Type C without IFN cases. Similarly to the case of ALB-PLT trajectories, we split the data into 15 clusters where dissimilarity increased largely at early stage. Table 1 provides cluster constitution stratified by fibrotic stage. In Table 1, we could observe a clear feature about the distribution of fibrotic stages over clusters. Clusters such as 3, 4, 6, 7 and 8 contained relatively large number of F3/F4 cases, whereas clusters such as 9, 11, 12, 13, 14, 15 contained no F3/F4 cases. These two types of clusters were divided at the second branch on the dendrogram;

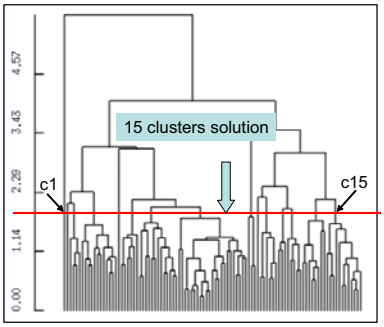

Figure 5: Dendrogram for Type C without IFN dataset (CHE-PLT trajectories).

Table 1: Cluster constitutions of CHE-PLT trajectories, stratified by fibrotic stages. Small clusters of $N < 2$ were omitted.

Cluster	# of Cases / Fibrotic stage				Total
	F0,F1	F2	F3	F4	
3	0	0	1	3	4
4	2	1	2	7	12
6	3	0	1	2	6
7	5	2	3	3	13
8	9	8	4	2	23
9	1	2	0	0	3
11	4	2	0	0	6
12	2	0	1	0	3
13	5	0	0	0	5
14	8	0	0	0	8
15	12	0	0	0	12

therefore it implied that, with respect to the similarity of trajectories, the data can be globally split into two categories, one contains the progressed cases and another contained un-progressed cases.

Now let us examine the features of trajectories grouped into each cluster. Figure 6 shows CHE-PLT trajectories grouped into cluster 3. The bottom part of the figure provides the legend. The horizontal axis corresponds to CHE, and the vertical axis corresponds to PLT. This cluster contained four cases: one F3 and three F4. The trajectories settled around the lower bounds of the normal range for PLT ($120 \times 10^3/ul$), and below the lower bounds of CHE (180 IU/l), with global direction toward lower values. This meant that, in these cases, CHE deviated from normal range earlier than PLT.

Figure 7 shows trajectories grouped into cluster 4, which contained nine F3/F4 cases and three other cases. Trajectories in this cluster exhibited interesting characteristics. First, they had very clear descending shapes; in contrast to trajectories in other clusters in which trajectories changed directions frequently and largely, they moved toward the left corner with little directional changes. Second, most of the trajectories settled below the normal bound of PLT whereas their CHE values ranged within normal range at early phase. This meant that, in these cases, CHE deviated from normal range later than PLT.

Figure 8 shows trajectories grouped into cluster 6, which contained three F3/F4 cases and three other cases. Trajectories in this cluster exhibited descending shapes similarly to the cases in cluster 4. The average levels of PLT were higher than those in cluster 4, and did not largely deviated from the normal range. CHE remained within the normal range for most of the observations.

Figure 9 shows trajectories grouped into cluster 15, which contained twelve F0/F1 cases and no other cases. In contrast to the high stage cases mentioned above, trajectories settled within the normal ranges for both CHE and PLT and did not exhibit any remarkable features about their directions.

These results suggested the followings about the CHE-PLT trajectories on type C without IFN cases used in this experiment: (1) They could be globally divided into two categories, one containing high-stage cases and another containing low-stage cases, (2) trajectories in some high-stage clusters exhibited very clear descending shapes. (3) in a group containing descending trajectories, PLT deviated from normal range faster than CHE, however, in another group containing descending trajectories, PLT deviated from normal range later than CHE.

4 Conclusions

In this paper we propose a trajectory clustering method as multivariate temporal data mining and shows its application to data on chronic hepatits. Our method consists of a two-stage approach. Firstly, it compares two trajectories based on their structural similarity and determines the best cor-

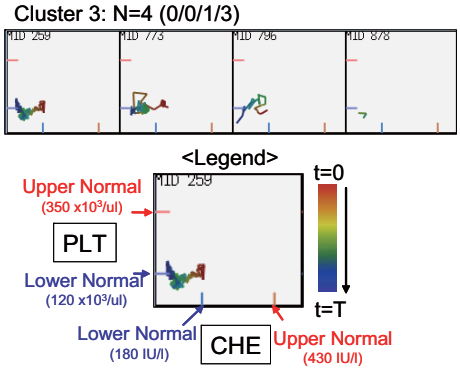

Figure 6: Trajectories in Cluster 3.

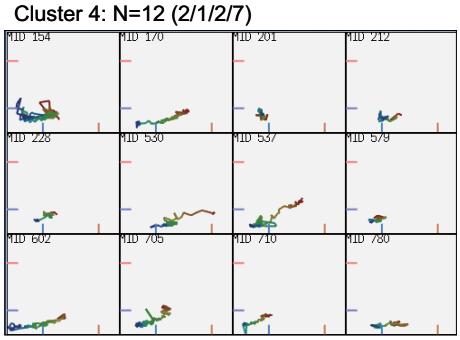

Figure 7: Trajectories in Cluster 4.

respondence of partial trajectories. Next, it calculates the value-based dissimilarity for the all pairs of matched segments and outputs the total sum as dissimilarity of the two trajectories.

Clustering experiments on the chronic hepatitis dataset yielded several interesting results. First, the clusters constructed with respect to the similarity of trajectories well matched with the distribution of fibrotic stages, especially with the distribution of high-stage cases and low-stage cases, for ALB-PLT, CHE-PLT and ALB-CHE trajectories. Among three combinations, ALB-CHE shows the highest degree of covariance, which means that CHE can be used to evaluate the trends of ALB.

Our next step is to extend bivariate trajectory analysis into multivariate one. From the viewpoint of medical application, our challenging issue will be to find a variable whose chronological trend is fitted to PLT.

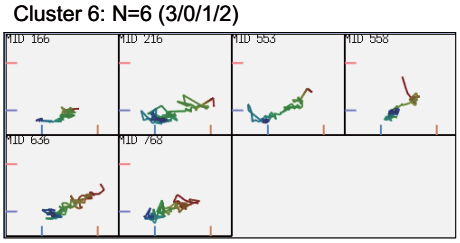

Figure 8: Trajectories in Cluster 6.

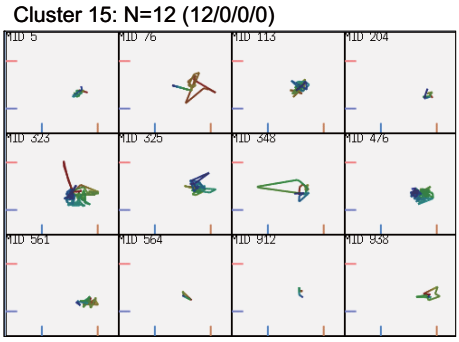

Figure 9: Trajectories in Cluster 15.

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.

[2] T. Lindeberg (1990): Scale-Space for Discrete Signals. IEEE Trans. PAMI, 12(3):234–254.

[3] B. S. Everitt, S. Landau, and M. Leese (2001): Cluster Analysis Fourth Edition. Arnold Publishers.

[4] S. Hirano and S. Tsumoto (2003): An Indiscernibility-Based Clustering Method with Iterative Refinement of Equivalence Relations - Rough Clustering -. Journal of Advanced Computational Intelligence and Intelligent Informatics, 7(2):169–177.

[5] URL: http://lisp.vse.cz/challenge/

[6] S. Tsumoto, S. Hirano, and K. Takabayashi (2005): Development of the Active Mining System in Medicine Based on Rough Sets. Journal of Japan Society of Artificial Intelligence, 20(2): 203–210.