Knowledge Discovery Based on Importance of Features

Hiroshi Sugimura
Graduate School of Engineering
Kanagawa Institute of Technology
Kanagawa, Japan
hiroshi.sugimura@gmail.com

Kazunori Matsumoto
Graduate School of Engineering
Kanagawa Institute of Technology
Kanagawa, Japan
matumoto@ic.kanagawa-it.ac.jp

***Abstract*—This paper proposes a system which datamines time series classification knowledge leading by a discovery of feature patterns. In the case of classification, prediction accuracy is an important point, and to build a human understandable model is another essential issue. To satisfy these requests, our system runs in two stages. In the first stage, the system discovers important feature patterns which are useful for identifying data. For this purpose, we propose a feature importance measure which is called FI. The second stage builds a decision tree that determines class membership based on the feature patterns. We explain how these two stages are harmonized in the entire process.**

Keywords-feature discovery; knowledge extraction; automatic improvement; classification; time series data

I. INTRODUCTION

Time series data consist of long sequences of numeric data, which are recorded at equal time intervals (e.g., per minute, per hour, or per day). This type of data can be generated by many natural and economic processes such as stock markets, and scientific, medical or natural observations [1]. For knowledge extraction method from such data is important.

In [2], they propose a method that efficiently discovers frequent patterns from time series data. This method regards the frequent patterns as feature patterns. However, the patterns do not always correspond to the interests of analysts, because the patterns are common and not necessarily a source of new knowledge for the analysts.

In [3], user specified subsequences of time series data are used as background knowledge. The method searches the space under constraint given in the regular expressions, and then extracts only subsequences that satisfy the regular expressions. Thus analysts can discover feature patterns that accord with their interests. However, this method depends on analysis experience. If the analysts have insufficient knowledge, the methods cannot discover feature patterns.

A classification method for time series data by using the support vector machine, SVM for short, is proposed in [4]. SVM strategy is a very powerful method that outperforms many other systems in a wide variety of applications. However, the classification model of SVM classifiers is non-understandable.

This paper proposes a method that runs without using background knowledge given by a user. Our method starts with a feature discovery process. In order to discover feature shapes, we propose a feature importance measure FI which reflects both global and local viewpoints. The main idea of FI is that a non-frequent pattern from a global viewpoint can become important feature if it occurs frequent in some local area. We explore the possible space of subsequences by considering the value of FI. The discovered feature is used as the attribute of the classification. For maintaining the understandability of a classified result, we use the decision tree learning. The decision tree learning can generate some rules to describe the decision model.

In addition, we further provide a method that improves discovered patterns by using the genetic algorithm (GA). GA is an adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetics [5]. This heuristic is generally used to generate useful solutions for optimization and search problems.

In [6], they propose a method to encode and decode a decision tree to and from a chromosome where genetic operators such as mutation and crossover can be applied are presented. Our approach repeatedly increments the quality of a decision tree by improving feature patterns. This paper demonstrates the effectiveness of the method to real data.

II. FEATURE PATTERN DISCOVERY

A. Preliminaries

A time series data S is a finite sequence of real values $S = (S_1, S_2, \cdots, S_n)$, where n is the length of this sequence. We in this paper assume each time series data is classified into predefined categories, where they are associated with class values.

As we explained above, the first task is a discovery of feature patterns from time series data. It is clear that a frequent pattern in a time series data is not negligible, yet if a pattern

978-1-4673-1686-6/12/$26.00 ©2012 IEEE

exists uniformly in the entire database its importance decreases. Fig. 1 shows this idea.

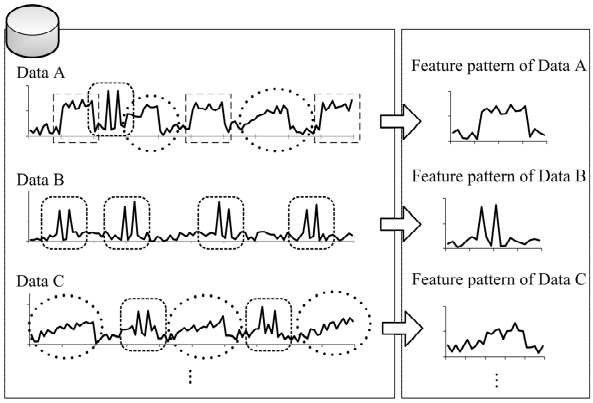


Figure 1. Feature pattern discovery

B. *Feature importance*

In order to discover feature subsequences from time series database, we propose an importance measure which is named FI. In the area of text mining, a similar idea is proposed and is called TF $*$ IDF [7]. This method deals with a set of documents, and a document is regarded as a simple set of terms. In other words, terms are the only tools to identify characteristics of documents. Thus we need a method to measure the importance of terms. We extend this idea over time series data.

The key point of our idea is to provide two viewpoints, local and global, in analyzing time series data. In the case of local viewpoint, we focus on frequently occurred patterns in one time series data and give high importance to them. On the other hand, the global viewpoint focuses on uniqueness of patterns over a collection of time series data. Thus we offset the importance of patterns that occur commonly over many time series data. The total importance of a pattern is calculated by taking these two aspects into consideration. To realize this idea, we divide a time series data α_i into subsequences $\alpha_i^1, \alpha_i^2, \alpha_i^3, \cdots$ by using slide-window method, and the number of the subsequences is denoted by $\#s(\alpha_i)$. We assume there are N time series data in a target database. For a given pattern p, which is an arbitrary subsequence, the number of subsequences of α_i that include p is denoted by $\#(p \wedge s(\alpha_i))$. We denote, by $\#m_p$, the number of time sequences whose subsequence include p. The importance of a pattern p inside a time series α_i is expressed by FI and is formally defined as follows:

$$\mathrm{FI}(p,\alpha_i) = \frac{\#(p \wedge s(\alpha_i))}{\#s(\alpha_i)} \times \log\left(\frac{N}{m_p}\right) \qquad (1)$$

The total number of subsequences becomes huge, therefore we find representatives of subsequences. We develop clustering techniques for this purpose that is explained more in detail in the following.

In order to discover representative subsequences, we apply the k-means clustering to extracted subsequences, which is one of the most popular clustering algorithms [8]. For a given data set, the k-clustering aims at partitioning this data set into k disjoint subsets (clusters), such that a clustering criterion is optimized. The clustering criterion here is to minimize the distance between each data v and the cluster center g_i (centroid) of the group G_i which contains v. This criterion is called clustering error, and depends on the centroids $g_1, \cdots g_k$, where $\mathrm{D}(P, Q)$ is a distance function between P and Q.

$$Err = \sum_{i=1}^{k} \sum_{v \in G_i} D(v, g_i) \qquad (2)$$

The algorithm runs in accordance with the following steps:

1. Choose the number of clusters k,
2. Randomly generate k clusters and determine the centroids,
3. Assign each point to the nearest centroid, where nearest is defined with respect to one of the distance measurements $\mathrm{D}(P, Q)$,
4. Recompute the new centroids, and
5. Repeat the two previous steps until some convergence criterion is met (usually that the assignment has not changed).

The most widely used distance function $\mathrm{D}(P, Q)$ is the Euclidean distance or its variations. However, Euclidean distance can be an extremely brittle distance measurement as we show in Fig. 2 (a). Euclidean distance may fail to produce an intuitively correct measurement of similarity between two sequences because it is very sensitive to small distortions in the time axis. For this reason, we use Dynamic Time Warping.

Dynamic Time Warping [3], DTW for short, is an algorithm for measuring distance between two time series data. A naive approach to calculating a matching distance between two time series is a repeated calculation of the difference of them at each time point. The drawback of this method is that it does not produce intuitive results.

DTW solves this discrepancy between intuition and calculated matching distance by recovering optimal alignments between points in the two time series. The alignment is optimal in the sense that it minimizes a cumulative distance measure consisting of local distances between aligned sequences. Fig. 2 (b) shows such an alignment. This is called Time Warping because it warps the time axes of the two time series in such a way that corresponding sequences appear at the same location on a common time axis.

For two sequences that have different lengths i, and j, the dissimilarity degree $f(p_i, q_j)$ is defined in the following equation, where p_{i-1} and q_{j-1} is the sequences with the last values removed.

In equation (3), s and t represent the cost of shortening and expanding the sequences along the time axis, and u is the distance cost of values. A more similar pair has a smaller value,

which becomes zero when they are exactly the same. For a given threshold value, we regard them as equal.

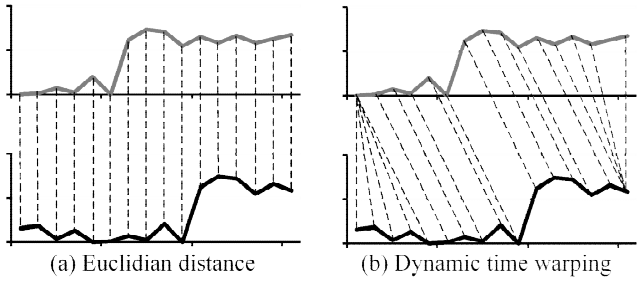


Figure 2. Euclidean distance and Dynamic time warping

$$f(p_i, q_j) = \min \begin{cases} f(p_i, q_{j-1}) + s \\ f(p_{i-1}, q_j) + t \\ f(p_{i-1}, q_{j-1}) + u \end{cases} \quad (3)$$

III. EXTRACTING CLASSIFICATION RULES

Extraction of classification rules is carried out by using the feature patterns that are identified in the previous task. The extraction goes as an iterated process that includes (1) candidate discovery, (2) evaluation, and (3) improvement of candidates. First, the classification rules are extracted by the decision tree learning on the basis of discovered feature patterns. Second, extracted rules are evaluated from the viewpoints of classification accuracy. When it exceeds predefined accuracy, terminates and putting the discovered rules. Third, in order to increase the effectiveness of the feature patterns they are automatically improved, and return to the first step. This process is repeated until a termination condition has been reached in the second step.

A. Making of training data based on feature patterns

Fig. 3 (a) illustrates an example of training data. From F_1 to F_3 are feature patterns that are discovered by feature discovery step. From S_1 to S_4 are subsequences in database. We compute the degree of similarity of training data and feature patterns. Each instance in the training data is associated with a class label. Thus table consists of tuples of these values and class labels. As a result, we obtain knowledge as shown in Fig. 3 (b).

B. Improvement of patterns

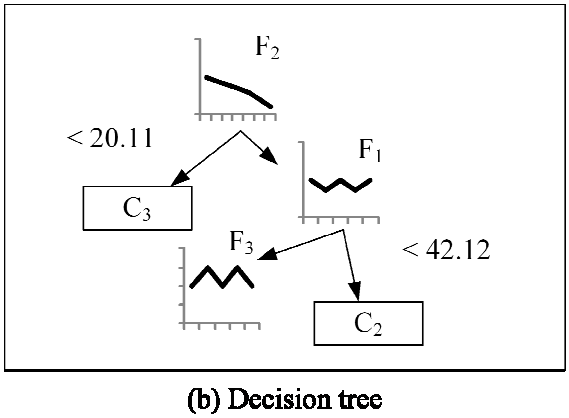
The quality of classification rules depends on the properties of the feature patterns. Thus, the improvement task carries out a reformulation of the feature patterns. In order to increment of the quality of knowledge, the local search such as hill climbing and the reinforcement learning such as Q-learning can be applied. However, it may obtain a local optimal solution. Our approach is to use the GA. GA is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems.

C. Gene representation

The representation of a feature pattern is an array of numerical values. Thus, we regard each pattern as each gene in GA. A set of current genes becomes current generation of a family.

Data	F_1	F_2	F_3	class
S_1	50.91	73.33	128.33	C_1
S_2	60.3	83.34	168.33	C_2
S_3	33.33	40.23	95.23	C_2
S_4	33.33	0	95.91	C_3

(a) Training data



(b) Decision tree

Figure 3. Example of training data and decision tree

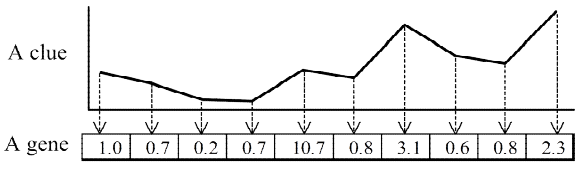


Figure 4. Gene representation

We explain a fitness function used here. The information gain ratio criterion is used to determine the most appropriate gene for classifying all instances of training data. As we show in Fig. 3, all instances are classified, and similarities are computed between their sequences and all genes. Intuitively fitness of gene is decided by the degree of contribution in the classification task. For this purpose, the system computes information gain to find an adequate fitness value.

In order to determine the information gain ratio, we have to look at the information conveyed by classified cases. We consider a set T of k training cases. If we select a single case t in T and decide that it belongs to class C_j, then the probability of this message is $\frac{\text{freq}(C_j,T)}{|T|}$ and it conveyed $-\log_2\left(\frac{\text{freq}(C_j,T)}{|T|}\right)$ bits of information. Then the average amount of information needed to identify the class of a case in set T can be computed as a weighted sum of per-case information amounts:

$$\text{info}(T) = -\sum_{j=1}^{u} \frac{\text{freq}(C_j,T)}{|T|} \times \log_2 \frac{\text{freq}(C_j,T)}{|T|} \qquad (4)$$

If the set T is partitioned into n subsets on the basis of outcomes of test X, we can compute a similar information requirement as:

$$\text{info}_x(X) = \sum_{i=1}^{w} \frac{|T_i|}{|T|} \times \text{info}(T_i) \qquad (5)$$

Then, the information gained by partitioning T in accordance with the test X can be computed as:

$$\text{gain}(X) = \text{info}(T) - \text{info}_x(T) \qquad (6)$$

The gain criterion is biased toward the high frequency data. To ameliorate this problem, we normalize the information gain by the amount of the potential information generated by dividing T into n subsets:

$$\text{split info}(X) = -\sum_{i=1}^{w} \frac{|T_i|}{|T|} \times \log_2 \frac{|T_i|}{|T|} \qquad (7)$$

Thus the gain ratio is defined as:

$$\text{gain ratio}(X) = \frac{\text{gain}(X)}{\text{splitinfo}(X)} \qquad (8)$$

D. Selection

Selection step rates the fitness of each gene and preferentially selects the best gene. We use roulette wheel selection [5] to do this. Typically, a proportion of the wheel is allocated to each of the possible selections on the basis of their fitness value. With fitness proportionate selection, some weaker genes may survive the selection process; this is an advantage, as although a gene may be weak, it may contain some components that could prove useful following the reproduction process. The roulette wheel selection algorithm consists of the following steps.

1. Sum the fitness of all population members; named total fitness, n,
2. Generate a random number between zero and n, and
3. Return the first population member whose fitness added to the fitness of the preceding population members is greater than or equal to n.

E. Reproduction

The reproduction step is to create a next generation population of genes from those selected through genetic operators that are crossovers and mutated. To create each new gene, a pair of parent genes is selected for breeding from the pool selected previously. By creating a child gene using these methods, a new gene is created that has many of its parents' characteristics. The process continues until a new population of genes of suitable size has been generated. The mutation step adds to the current value of a gene with a randomly generated valid value that is in the range *-50* to *+50*.

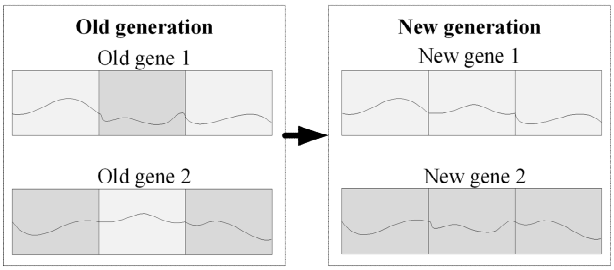


Figure 5. Cross over

IV. EXPERIMENT

In order to confirm the effectiveness of our proposed system, the experiments use real hepatitis dataset, which is produced by the Chiba University hospital, and is distributed at the PKDD Discovery Challenge 2004 and 2005. This contains long time series data on the laboratory examinations of hepatitis B and C infected patients. The examinations are realized between 1982 and 2001 on 771 patients. We aim to discover temporal differences between the hepatitis B and C from platelet count data. Platelet (PLT) count has been receiving considerable interests as an index for liver dysfunctions, because a hematogenetic factor called thrombopoietin [10], which facilitates the production of platelets, is produced in the liver.

In Fig. 6 we show an outline of this experiment. A medical examination record of one patient is expressed as a sequence of time series data. For each record, we first collect, by sliding window, the set of all subsequences having predetermined length. We carry out a clustering process on the whole subsequences one the entire records. In the figure, the R_i is the representative subsequence of each cluster. In the next step, some representations are filtered out according to the TF $*$ IDF measure, and then survived ones become feature patterns. Each patient record is re-expressed with a use of the feature patterns. This becomes a training data in the classification rule extraction that uses decision tree learning. We remark here each record is classified into three types of hepatitis.

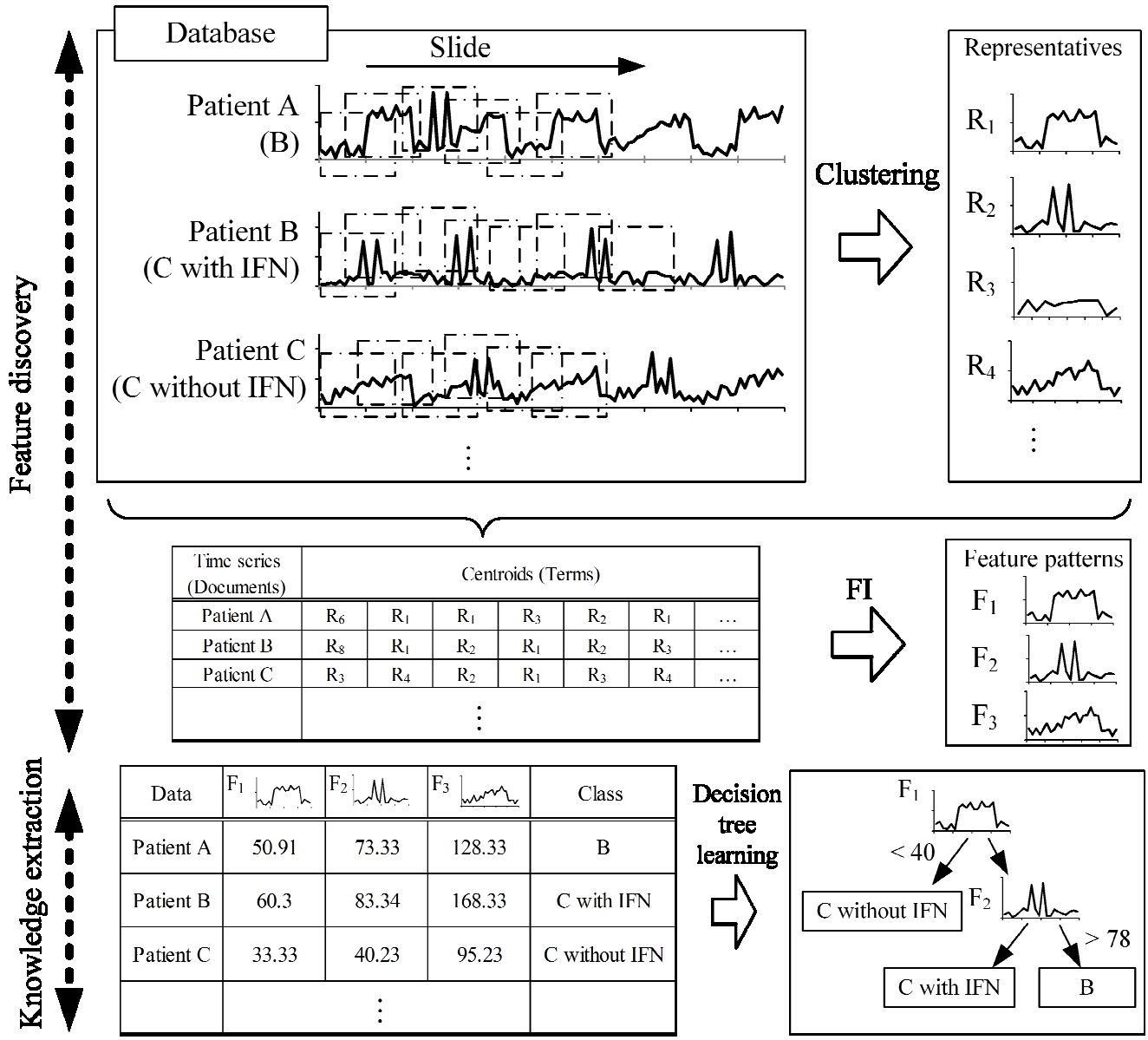


Figure 6. An outline of the system

The slide window length is *20* data, which is one month. We use the C4.5 algorithm [11] within the WEKA [12], which is developed at the University of Waikato in New Zealand. All decision tree setting is some. For pruning of C4.5, confidence factor is *0.25*, and minimum number of the objects for leaf node is *2*. In the DTW, the cost of horizontal (time axis) is *50.0*, and the cost of vertical is difference of values $|p_i - q_i|$.

In the experiment for the improvement of patterns, genetic algorithm increases the population of patterns one hundred times. The system shows the patterns and the decision tree that has the highest accuracy. When accuracy is the same, we prefer smaller trees. To evaluate the prediction accuracy, we use the 10-fold cross validation. In order to compare accuracy, we carry out other three methods as following.

- Decision Tree Learning (DTL for short)
- Neural Network (NN for short)
- Support Vector Machine (SVM for short)
- Our approach (10, 20, and 30 clusters)

Methods DTL, NN and SVM deal with subsequences by using sliding window. Our approach uses the feature patterns by the feature discovery as an initial gene. The experiments are carried out using three sets of genes which contain total numbers of gene are ten, twenty and thirty. These genes are

selected in descending order of value of *FI*. Table 1 shows these results of accuracy.

DTL uses C4.5 which has properties as above. NN uses Multilayer Perceptron which has properties as follows: training time is *500*, number of hidden layers is one, number of neuron in hidden layers is *20*, learning rate is *0.3*, and momentum term is *0.2*. SVM uses the Sequential Minimal Optimization which uses the poly kernel.

TABLE I. ACCURACY COMPARISON

Method	Accuracy
DTL	67.37 %
NN	54.48 %
SVM	54.37 %
cluster k = 10	74.03 %
cluster k = 20	79.18 %
cluster k = 30	80.68 %

Fig. 7 shows number of cluster k and total number of discovered feature patterns. The feature patterns are the clusters which have the maximum value of *FI* of each data. Same cluster is extracted from similar time series data. Fig. 8 shows graph of the accuracy obtained by each generation with GA.

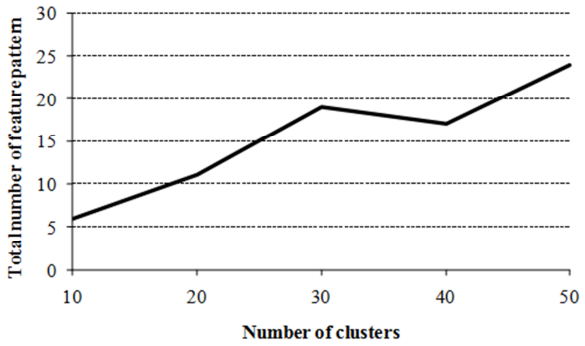


Figure 7. Total number of clusters

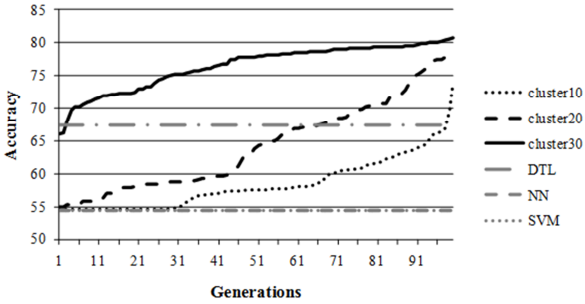


Figure 8. Improvement by GA

V. DISCUSSION

As we can see in Fig. 7, in accordance with an increase of the number of clusters, the total number of discovered feature patterns is increased, and the accuracy of the classifier is also increased. In Table 1, the accuracy of our approaches is greater than other method. In addition, the accuracy of our approach is improved by using GA. Fig. 8 shows the accuracy is improvement generation by generation.

VI. CONCLUSION

This paper proposes a method that runs without using background knowledge given by a user. Our method starts with feature discovery process. In order to discover feature shapes, we propose a feature importance *FI* which reflects statistical measurements and it importance. The main idea of *FI* is similar to the TF $*$ IDF weight, which is originally invented in text mining. We explore the possible space of subsequences by considering the value of *FI*. The discovered feature is used to the attribute of the classification. For maintaining the understandability of a classified result, we use the decision tree learning. The decision tree learning can generate some rules to describe the decision model.

In addition, we further provide a method that improves discovered patterns by using the GA. The improvement function is effective when the initial accuracy of decision model is low.

REFERENCES

[1] Jiewei Han, Micheline Kamber and Jian Pei, "Data mining: concepts and techniques", *Morgan Kaufmann*, 2011.

[2] R. Agrawal and R. Srikant, "Mining sequential patterns," *in Data Engineering, 1995. In Proceedings of the Eleventh International Conference on*, pp. 3-14, IEEE, 2002.

[3] K. Haigh, W. Foslien and V. Guralnik, "Visual Query Language: Finding patterns in and relationships among time series data," *in Proceedings of the seventh Workshop on Mining Scientific and Engineering*, DatasetsCiteseer, 2004.

[4] A. Kampouraki, G. Manis and C. Nikou, "Heartbeat time series classification with support vector machines," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 13, No. 4, pp. 512-518, 2009.

[5] J.H. Holland, "Adaptation in Natural and Artificial Systems," *University of Michigan Press*, 1975.

[6] S. Hyuk Cha and C. Tappert, "A genetic algorithm for constructing compact binary decision trees," *in Journal of Pattern Recognition Research (JPRR)*, vol. 4, No. 1, pp. 1-13, 2009.

[7] K.S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, No. 1, pp. 11-21, 1972.

[8] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman and A. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, No. 7, pp. 881-892, 2002.

[9] D.J. Berndt and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series," *in Proceedings of KDD-94: AAAI Workshop on Knowledge Discovery in Databases*, pp. 359-370. Seattle, Washington, 1994.

[10] H. Miyazaki, "Future Prospect of Thrombopoietin," *Jpn J. Transfusion Medicine,* vol. 46, No. 3, pp. 311-316, 2000.

[11] J.R. Quinlan, "C4.5: Programs for Machine Learning," *Morgan Kaufmann Publishers*, 1993.

[12] The University of Waikato, Machine learning project at the university of waikato in new zealand, http://www.cs.waikato.ac.nz/ml/, 2009.