

Clues Driven Time Series Data Mining with an Automatic Improvement Mechanism

Hiroshi SUGIMURA
*Graduate School of Engineering
 Kanagawa Institute of Technology
 Kanagawa, Japan*

Kazunori MATSUMOTO
*Graduate School of Engineering
 Kanagawa Institute of Technology
 Kanagawa, Japan*

Abstract—This paper proposes a data mining system that acquires knowledge from time series data by using clues. To obtain if-then rules as knowledge, we apply decision tree learning. However, in decision tree learning, a simple method that regards a value at each point as an attribute does not deal with the proper shape of time series data. It also problematically makes decision trees that become large and complex. We focus on the human behavior. Experts forecast future events by using their knowledge. They, in typical cases, focus on a set of useful patterns and then apply knowledge relevant to them. We apply this idea into the framework of decision tree learning. We prepare a set of patterns, which is called clues, and then express time series data in terms of the clues. Thus the clues are attributes by which features of time series data are described. To make a better prediction with the learning process, we develop a mechanism that improves the quality of the clues. The essential idea of the mechanism is based on a genetic algorithm. The clue is evaluated by using entropy of information theory, and is improved by GA operators. We can obtain new knowledge from improved clues and the extracted decision tree. This paper details the system and results of the experiment.

Keywords—datamining; time series; dynamic time warping; decision tree; genetic algorithm

I. INTRODUCTION

Time series data occur frequently in business applications and in science. Typical examples include daily stock prices on the Tokyo stock market, hourly TV audience ratings, and monthly sea surface temperatures in the Equatorial Pacific. Time series data are typically plotted via line charts. Our study focuses on extracting knowledge from such data.

The purpose of data mining is to extract implicit, previously unknown, and potentially useful information from data. Its process must be automatic or semiautomatic. The patterns discovered must be meaningful in that they lead to some advantage, usually an economic advantage. The data is invariably present in substantial quantities. Artificial Neural Networks can forecast time series data relatively well but cannot explain the forecasting rules clearly. On the other hand, the decision tree learning can generate some rules to describe the forecasting decisions.

This study carries out extraction knowledge from time series data using decision tree learning. The decision tree learning has an advantage in that it easily represents knowledge to an if-then rule, and a lot of successful cases have

been reported [1], [2]. However, the node of typical decision tree learning cannot deal with attributes of time series data. We thus need to preprocess data in this case. A simple method is to rearrange time series data to its average. However, this disregards the shape of time series data, thus two sets of data may be incorrectly deemed to be similar even if they have greatly different shapes. Time series has many more features that can be taken into account for it to be a potential candidate for the method that works best.

In [3], they develop a method that acquires knowledge by using decision tree learning. This method considers data sets that consist of two or more time series attributes. On the other hand, we aim to discover the knowledge from one-long-period time series data. In [4], a method to predict future behavior by using clustering is proposed. However, several studies point out that brute force exploration often identifies meaningless simple behavior like a cosine curve [5].

To avoid this problem we need to investigate patterns appearing in time series. A useful and non-trivial pattern is hard to identify automatically. Thus, we use human experience, which is expressed in clues, as hints of useful and non-trivial patterns. They become features on attributes in the learning. In another characteristic of time series data, we need a method to manage ambiguity in data. We need to identify sets of data that have similar shapes, different lengths, different values at some points, etc. For this propose, we use a technology developed in image processing.

In addition, we describe a method that improves clues by using the genetic algorithm, GA for short. GA is an adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetics [6]. This heuristic is generally used to generate useful solutions for optimization and search problems. In [7], 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 indirectly increments the quality of a decision tree by improving of clues. This paper also outlines the entire system and the methodology.

II. TIME SERIES DATA MINING BY USING CLUES

Time series data consist of a set of time sequences, each of which represents values sorted in chronological order,

and are abundant in various domains. We predict the future behavior of time series data by using classification methods. Typical classification methods can be classified into transformation and direct approaches. The former maps a time sequence to another representation as the median. On the other hand, the latter deals with n values on n time points as n attribute values. These methods neglect the characteristics of a time sequence, and might possibly recognize a pair of largely different time sequences as similar. Our approach solves this problem by using clues.

A. Use of clues

Much empirical knowledge is in the financial areas, especially stock price analysis and prediction. In this area, typical knowledge is used to predict future price changes on the basis of examinations of past price changes. Most knowledge in this area is described on the basis of charts and patterns. A pattern is a subsequence of time series data and identifies a distinctive behavior of the sequence. In many cases, patterns are effectively used to represent knowledge of time series data. The existing knowledge of patterns plays an important role in our discovery process. We call pattern a “clue”.

This method has two advantages. Firstly, system can use the user’s knowledge. When the user knows previously effective patterns, knowledge can be efficiently acquired by input patterns. Secondly, the system can discover a set of superior clues. If the user does not know any previously effective patterns, the system randomly generates the initial clues, and discovers useful knowledge by improving these clues. A user can obtain improved clues as new knowledge.

B. Outline of the system

The system consists of the following steps.

- 1) Data collection
- 2) Preprocessing
- 3) Knowledge discovery
- 4) Evaluation
- 5) Improvement of clue

First, a user inputs and selects target data: clues, raw databases, and real-time data. The target data is then cleaned. Cleaning removes the observations that have noise and missing data. Initial clues are also given. Second, the system carries out a preprocessing operation for common processes. The collected data come in various types, we thus need to normalize them. Third, the knowledge is discovered by the machine learning with clues. Fourth, discovered knowledge is evaluated. In accordance with this evaluation, the system chooses “stop” or “improvement”. If it chooses “stop”, the system outputs the result that is a set of discovered knowledge and used clues. When it chooses “improvement”, the system carries out step five. Fifth, the clues are automatically improved on the basis of the evaluation, and return to the third step. This process is repeated until a termination

condition has been reached in the fourth step. Next, we describe the data mining method based on clues and also the method for improving clues.

C. Training data

The user puts several sequences of values into the system as clues. The input sequences are normalized as follows, and stored in a database. We consider a sequence of a clue $C = I_1, \dots, I_n$. Each value of a sequence of a normalized clue is obtained by I_n/I_{n-1} . The first value I_1 is 1.00.

Several time series data are retrieved and picked up from a database. These time series data are classified by their future behavior. The best match parts that are most similar to a clue from each data are retrieved. Dissimilarities between a clue and best match parts are computed. The system applies this operation to all clues.

Fig. 1 shows a simple example of the method that makes an instance in the training data from time series data. Clues are given to a data mining processor and used as ‘focus points’ in the mining algorithm. As a result, we obtain knowledge as shown in Fig. 2.

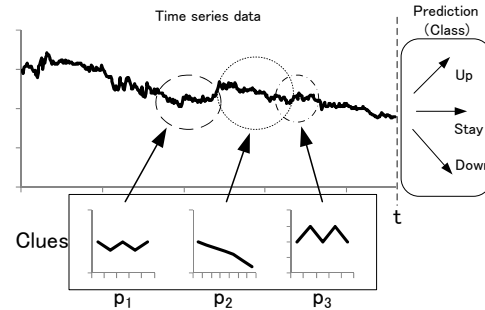


Figure 1. Simple example of training

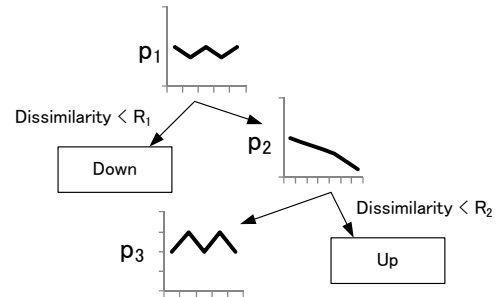


Figure 2. An example of result

Fig. 3 illustrates an example of training data. Clues 1 to 4 is the given set of clues in this example. For all training data and each clue, we compute how well they match. Each instance in the training data is associated with a class label, which denotes future behavior. A table that is tuples of these dissimilarities and classes is training data.










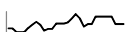

Sequence Data	clue 1	clue 2	clue 3	clue 4	clue 5	clue 6	clue 7	class {up, down, stay}
0-19 								→ stay
5-24 	50	120	128.34	83.33	140	0	121.67	→ stay
10-29 	33.33	103.33	95	40	100	325	121.67	↘ down
15-34 	33.33	103.33	95	0	100	325	171.67	↗ up

Figure 3. Training data by using clue

We point out a crucial issue with dealing with clues. Clues are taken from existing knowledge, and they define abstract behavior of time series data. Thus, actual data seldom match the clues. First, this clue may correspond to short or long period behavior. Thus we can say there is horizontal ambiguity. Similarly, there is vertical ambiguity that handles differences of values.

D. Pattern matching

Typically Euclidean distance is varied or extended. However, Euclidean distance can be a very brittle distance measure. 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. Fig. 4 (a) shows correspondence point pairs by using Euclidean distance. The two sequences have approximately the same component behavior, but this behavior does not line up in time axis. Fig. 4 (b) shows the nonlinear alignment that allows a more sophisticated distance measurement to be calculated.

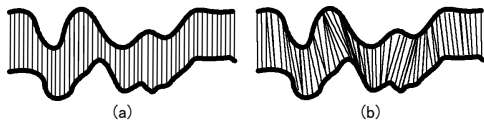


Figure 4. Euclidean distance and DTW distance

For these ambiguities, we apply the dynamic time warping [8], DTW for short. DTW computes dissimilarity of two sequences by using predefined cost parameters. For two sequences that have different lengths i , and j , the dissimilarity degree $g(i, j)$ is defined in the following equation, where $i-1$ and $j-1$ is the sequences with the last values removed.

$$g(i, j) = \min \begin{cases} g(i, j-1) + q \\ g(i-1, j) + r \\ g(i-1, j-1) + s \end{cases} \quad (1)$$

In the equation (1), q and r represent the cost of shortening and expanding the sequences along the time axis, and s 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.

III. IMPROVEMENT OF CLUES

We also describe the mechanism that increments the quality of knowledge. To increment the quality of knowledge, the local search (hill climbing, for example) and the reinforcement learning (Q-learning, for example) can be applied. However, it may obtain a local optimal solution. Our solution is to use the genetic algorithm.

A. Genetic algorithm

As we described above, a set of knowledge works as clues of knowledge discovery. We naturally expect better clues to discover better knowledge. Thus, the quality of knowledge depends on the quality of clues. The system improves the decision tree by improving clues by using GA.

B. Gene representation

The representation of a clue is an array of numerical values. Each clue becomes each gene by normalization. A set of current genes becomes current generation of a family. Fig. 5 shows representation of a gene that corresponds to a clue.

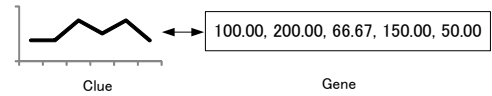


Figure 5. A clue and a gene

C. Fitness function

The system utilizes the information gain ratio criterion as evaluation of genes. The information gain ratio criterion is used to determine the most appropriate gene for classifying all instances of training data. As we describe in Fig. 3, all instances are classified by future behavior, and similarities are computed between their sequences and all genes. The system computes information gain for each gene by using training data.

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 conveys $-\log_2 \frac{\text{freq}(C_j, T)}{|T|}$ 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:

$$\text{info}(T) = - \sum_{j=1}^k \frac{\text{freq}(C_j, T)}{|T|} \times \log_2 \frac{\text{freq}(C_j, T)}{|T|} \quad (2)$$

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

$$\text{info}_X(T) = \sum_{i=1}^n \frac{|T_i|}{|T|} \times \text{info}(T_i) \quad (3)$$

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

$$\text{gain}(X) = \text{info}(T) - \text{info}_X(T) \quad (4)$$

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:

$$\text{split info}(X) = - \sum_{i=1}^n \frac{|T_i|}{|T|} \times \log_2 \frac{|T_i|}{|T|} \quad (5)$$

Thus the gain ratio is defined as:

$$\text{gain ratio}(X) = \frac{\text{gain}(X)}{\text{split info}(X)} \quad (6)$$

D. Selection

Selection step rates the fitness of each gene and preferentially selects the best gene. We use roulette wheel selection [6] to do this. A proportion of the wheel is typically 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 0 and n .
- 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 is 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.

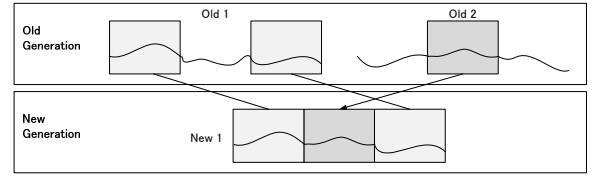


Figure 6. Two-points crossover

IV. EXPERIMENT

To confirm the effectiveness of our proposed system, the experiments use virtual data and real data. The virtual data are idealized data for the proposed system. The real data are two kinds: stock price data and climate data.

Time series data are extracted by using the slide window method. The slide window length is 20 data. Observation period of the future data is the next 5 data. In accordance with behavior of the period, time series data is classified. We consider three classes: up, down, and stay. Let x be the rate of stay and l be the final value of time series data. If any value in the future observation period is equal or larger than $l \times (1 + x)$, the class is up. Similarly, the class is down if any value in the period is equal or smaller than $l \times (1 - x)$. When the class is neither up nor down, it is stay.

This rate x is determined by a prior experiment to reduce the class distribution bias of the dataset. This is because a very unbalanced dataset will have an overall accuracy depending on the bias. We use the C4.5 algorithm [9] within the WEKA [10], which is developed at the University of Waikato in New Zealand.

In the experimental for the improvement of clues, GA increases the population of clues one hundred times. The system shows the clues 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. The variance is calculated to confirm the stability of accuracy.

A. Virtual data

To make the virtual data sets, we prepare beforehand a set of several patterns that have priority and future behavior.

This set is called a knowledge set. We also prepare a set of several random patterns. The virtual data is made by aligning patterns in both sets at random. Fig. 7 summarizes this operation.

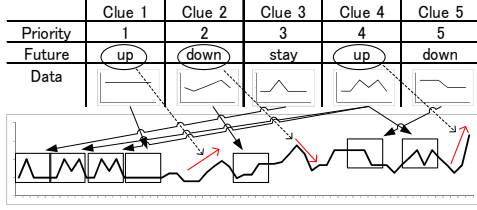


Figure 7. Making of virtual data

To compare the techniques, we carry out three kinds of experiments.

- Direct approach
- Using a knowledge set
- Using a random set

We experiment on 20 virtual data sets. Rate of stay x is 0.05. In the case of the system using clues, we measure the accuracy with various populations. Table I shows the average accuracy, the variance of accuracy, and the average size of a decision tree.

Table I
EXPERIMENTAL RESULTS OF VIRTUAL DATA

		Accuracy	Variance	Size
Direct		69.16	38.67	16.50
Random clues	5 genes	65.63	5.8	48.83
	10 genes	71.02	9.9	44.39
	20 genes	72.34	14.7	34.44
GA	5 genes	68.11	13.1	32.52
Random clues	10 genes	72.02	8.8	35.64
	20 genes	73.31	14.9	48.01
Known clues	5 genes	71.80	8.4	46.48
	10 genes	75.16	15.0	35.35
	20 genes	77.99	16.2	22.40
GA Known Clues	5 genes	79.87	17.0	16.60
	10 genes	82.29	11.2	20.00
	20 genes	84.27	4.4	22.68

B. Stock price data

We compare the accuracy and the stability of three kinds of methods.

- Direct approach
- "Hidenao Abe" method
- Our approach (using random set)

The time series stock data considered are row data from the Trade Science Corporation obtained from the Kaburobo website [11]. The data consisted of the opening price, the closing price, the highest price, and the lowest price of the trading days for one year. About 500 values are selected and used for experiment. The direct approach and our method generally take the closing price as the feature set for the

machine learning, and the prediction is created using the closing price only. Our approach generates ten random numbers from 50 to 150 for each clue, such as initial genes. Rate of stay x is 0.03.

The "Hidenao Abe" method is explained in section one. In accordance with the literature, using Kaburobo SDK, we obtained four price values, trading volume, and 13 trend index values (that are Moving Average, Bolinger Band, Envelope, HLband MACD, DMI, volume ratio, RSI, Momentum, Ichimoku1, Ichimoku2, Ichimoku3, and Ichimoku4).

We obtained time series data consisting of the above mentioned attributes about twenty five companies. The period, from we have collected the time series stock data, is from 5th January 2006 to 29st December 2006.

Table II shows experimental results of stock price data. The size of the tree and the accuracy are the average of twenty five companies. Fig. 8 shows the graph of the accuracy obtained by each approach.

Table II
EXPERIMENTAL RESULTS OF STOCK PRICE DATA

Method	Size	Accuracy	Variance
Direct	22.52	69.15	97.69
Abe	14.92	70.32	80.84
5genes	20.68	60.47	66.29
GA: 5genes	28.96	67.05	20.06
10genes	34.44	67.09	10.49
GA: 10genes	37.45	70.82	9.97
20genes	33.48	67.00	14.27
GA: 20genes	36.35	72.29	6.76

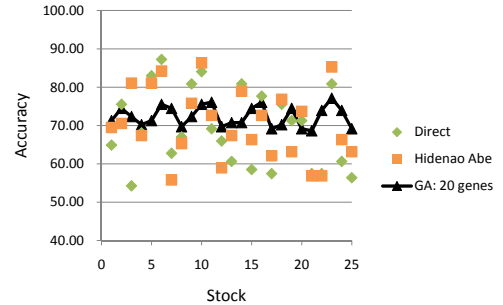


Figure 8. Accuracy obtained by each approach

C. Climate data

We use daily data of temperature and atmosphere. Both data set totals for 10 stations in Japan are obtained from the Global Historical Climate Network (GHCN) [12] for 50 years from 1st June 1959 to 31th May 2009. To consider climate change, we remove the trend from the data.

Table III shows results of the temperature, and Table IV shows results of the atmosphere.

V. DISCUSSION

In Fig. 8, the prediction accuracy of our approach is higher than other methods, and its stability is also good.

Table III
RESULTS OF THE ATMOSPHERE

Method	Size	Accuracy	Variance	Time(m)
Direct	5.2	99.79	0.01	0.15
5 genes	0.6	99.77	0.00	2.15
GA: 5 genes	2.4	99.78	0.00	7.50
10 genes	2.6	99.79	0.00	2.35
GA: 10 genes	5.2	99.81	0.00	14.50
20 genes	2.6	99.79	0.00	2.83
GA: 20 genes	5.0	99.83	0.00	28.30

Table IV
RESULTS OF THE TEMPERATURE

Method	Size	Accuracy	Variance	Time(m)
Direct	329.6	85.48	4.04	0.15
5 genes	15.8	53.84	0.45	2.15
GA: 5 genes	31.2	54.80	0.76	19.30
10 genes	58.2	54.66	0.83	4.00
GA: 10 genes	91.2	56.52	0.49	40.30
20 genes	72.6	55.06	1.08	6.10
GA: 20 genes	169.0	57.78	1.08	61.90

On the other hand, in Table IV, the direct approach obtains higher prediction accuracy than the proposed method over the data of simple behavior. We think that we should apply the proposed method to the data that changes by percentage, such as stock price data, rather than simple data, such as temperature data.

Table III and IV show that our approach takes longer for processing than the direct approach. The reason that is the decision tree is made many times by GA. In the experiment in this paper, the termination condition is when the 100th generation is calculated. However, it is thought that the processing time can be reduced by improving the termination condition. For example, we set the termination condition that is an accuracy threshold of a high value, such as 80%. It seems that we should apply our method to predict the long period rather than real time.

Next, we compare before and after improvement. In Table I, when five known clues are given, the accuracy improves by 8.07% (from 71.8% to 79.9%). When five random clues are given, the accuracy improves by 2.48% (from 65.63% to 68.11%). Table II shows the improvement in accuracy, but, Tables III and IV do not. These results, demonstrate that the GA-based mechanism enhances the prediction accuracy.

VI. CONCLUSION

This paper described a system that mines data for knowledge to predict future behavior of time series data. The main

point of the system is to use clues which are suggestive patterns for analysis. Clues become features on attributes in the learning, and are improved by using the genetic algorithm (GA). The experimental results demonstrated the effectiveness of the system.

REFERENCES

- [1] J. Pješivac-Grbović, G. Bosilca, G. Fagg, T. Angskun, and J. Dongarra, "Decision trees and MPI collective algorithm selection problem," *Euro-Par 2007 Parallel Processing*, pp. 107–117, 2007.
- [2] K. Kubota, A. Nakase, H. Sakai, and S. Oyanagi, "Parallelization of decision tree algorithm and its performance evaluation," *IPSJ SIG Notes*, vol. 99, no. 66, pp. 161–166, 1999. [Online]. Available: <http://ci.nii.ac.jp/naid/110002932396/en/>
- [3] Y. Yamada, E. Suzuki, H. Yokoi, and K. Takabayashi, "Decision-tree induction from time-series data based on a standard-example split test," in *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, 2003, pp. 840–847.
- [4] H. Abe, S. Hirabayashi, M. Ohsaki, and T. Yamaguchi, "Evaluating a trading rule mining method based on temporal pattern extraction," in *The Third International Workshop on Mining Complex Data (MCD2007) In Conjunction with ECML/PKDD 2007*, 2007, pp. 49–58.
- [5] E. Keogh, J. Lin, and W. Truppel, "Clustering of time series subsequences is meaningless: Implications for previous and future research," in *Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03)*, 2003, pp. 115–122.
- [6] J. H. Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [7] 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.
- [8] 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*, Seattle, Washington, Jul. 1994, pp. 359–370.
- [9] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.
- [10] T. U. of Waikato, "Machine learning project at the university of waikato in new zealand," <http://www.cs.waikato.ac.nz/ml/>, 2009.
- [11] KabuRobo, in Japanese. [Online]. Available: <http://www.kaburobo.jp>
- [12] N. E. Satellite, Data, and I. Service, "Nndc climate data online," <http://www.nesdis.noaa.gov/>, 2009. [Online]. Available: <http://www.nesdis.noaa.gov/>