九州大学学術情報リポジトリ
Kyushu University Institutional Repository

Syntactical Recognition Systems of Time Series based on the Genetic Programming and its Applications to Clustering of Stock Prices

陳, 暁栄
Antai School of Manegement, Shanghai Jiaotong University

時永, 祥三
九州大学大学院経済学研究院

https://doi.org/10.15017/7620

出版情報：經濟學研究. 71 (4), pp.107-119, 2005-07-29. 九州大学経済学会
バージョン：published
権利関係：

KYUSHU UNIVERSITY

Syntactical Recognition Systems of Time Series based on the Genetic Programming and its Applications to Clustering of Stock Prices

Xiaorong Chen and Shozo Tokinaga

1 Introduction

Advances in database management systems enable us to find some relationship among data in a systematic manner, and to use information retrievals on the basis of features of objects as well as conventional keys [1]. Similar situations are found in the database which is managing time series (temporal database) [1]. For example, many investment decisions are usually supported by the database system which maintains financial time series such as stock prices and exchange rates. Traders expect a rise of stock price if some shapes of ' triangle ' are found in the stock prices. For these management systems, a kind of recognition and clustering scheme is necessary by using numerical methods or syntactical methods [2]-[6].

This paper deals with clustering of segments of stock prices by using the syntactical recognition for time series based on the Genetic Programming (GP) [7][8]. We apply the GP procedure in learning phase of the system where we improve the functional forms (syntactical expressions) to approximate the models used to generate time series [9]-[20].

Conventional methods for clustering time series are categorized into two groups, and the first group contains various methods using the approximation system based on the basic functions, and the second group contains syntactical recognition methods [1][2]. For example, in the first group, the Fourie Transform and the Radial Basis Functions are used to expand the time series into the components, and then we examine these components to find features of the time series [1]. However, in these methods using the expansion into components we can not handle direct features in the time domain, and it is hard to find relations among temporal data.

The second method is resemble to the system for understanding natural language where the features of time series are described according to a kind of syntax so that we read and understand the meaning of the time series [2]-[6]. However, usually the subsystem to recognize the features of the time series are not simple, and the comprehensive building of the whole system becomes a tough task.

The GP procedure has been successfully applied to the estimation of chaotic dynamics using the observed time series, and a direct control method for chaotic dynamics is proposed based on the GP [9]-[15]. Moreover, the GP method has been widely used to emulate the agents ' behavior in various markets such as the stock market [14]. In this paper we extend the GP method to estimate and approximate syntactical expressions to fit the segments of time series, and its application to clustering of time series.

The GP method is effective in clustering phase as well as in learning phase. We assume that the time series is approximated by some arithmetic operations to basic patterns. These syntactical expressions are represented as tree structure (called individuals), and one tree corresponds to a model to generate time series. We have many individuals (pool) for the recognition

whether a time series belongs to a certain cluster. The individuals are improved by using the GP procedure in learning phase so that the estimation becomes to be better. The scheme to maintain the pool of individuals is necessary for the GP procedure, but it also contributes to absorb the variation of the time series in clusters. Because, it is reasonable to understand that syntactical expressions to approximate the time series in a cluster is not a single form while the member of the cluster are regarded to be generated from the original time series by expanding or shrinking the time scale. Then, the variation of the individuals with relatively high capability in the pool can cope with clustering for various kinds of time series which are seemed to belong to the same cluster. The scheme is the same as the classifier systems used in the GA-based optimization where a set of strings of binary data are used repeatedly depending on the change of environment [21]. If we use a single functional form to classify underlying time series, we may fail to approximate the time series.

As an application, we show clustering of artificially generated time series where clusters are obtained by expanding or shrinking by a transformation function. Then, we apply the system to clustering of 8 kinds of segments of real stock prices which are typically found in the technical analysis.

In the following, in Section 2, we describe modeling of time series and the overview of the system. In Section 3, we explain the basics of the GP procedure. Section 4 describes the application to clustering of artificially generated time series, and Section 5 shows the result of clustering of segment of real stock prices.

2 Description of Feature of Time Series

2.1 Clustering of time series using features

In conventional studies, the clustering of time series is usually formalized based on the knowledge of expertise having sufficient experiences. For example, in the medical diagnosis expertise compare the underlying EEG (Electroencephalogram) or ECG (Electrocardiogram) with the dataset of times series which are accumulated corresponding to each disease. It is also known in the technical analysis of stock trends, traders recognize the features of segments of stock prices so that they predict the rise or fall of price. Various patterns of stock trends are accumulated on the basis of specific features which are regarded to suggest us the rise/fall of stock price.

However, these human-based procedure are seemed to be costly and time consuming, then the automatization of the recognition process and the clustering of the time series may bring us the way of efficient investment. Moreover, it is also expected that the system can provide us the retrieval system of time series based on the feature of time series in relatively large database.

We must note following problems are needed to be resolved in clustering the time series based on the feature.

(1)range of data

Depending on the time and situation in observing the time series, the amplitude of the time series are not necessarily restricted in a small range. Therefore, some kind of function is needed for adjusting the variability of range of time series.

(2)Expansion and shrinking of time scale

It may also happen that the length of continuation of observation of times series or the segments of time series embedded with features are not the same in any instance. We may classify a group of time series to the same cluster where each time series has a similar figure but it is deformed after the figure is expanded or shrunk along the time. The problem of expansion and shrinking of time scale is usually resolved by using the dynamic programming

methods which are usually used in the recognition of voice. However, we may use another simpler method to treat the automatic clustering of conventional time series.
(3)Simplicity of feature description

It is also necessary to find simplified description of feature of time series so that we apply the various kinds of patterns of stock trends. Moreover, if the feature description is able to be a simple one, we can use the description also for the key in the retrieval of time series database.

The feature description and clustering method proposed in the paper based on the GP is expected to resolve above problems. For the first problem, the GP method improves the functional form for describing the relation among time series data, then difficulty caused by the difference of range occurred in each time series is almost always removed.

In term of the second problem, the functional form to describe the feature of time series is not a single form, but a pool of functional form is stored in the GP system. Then the variation and various patterns are realized in the same pool of individuals (functional forms) which are regarded to belong to the same cluster of time series.

For the simplified description of features (the third problem), we can control the complexity of the functional form in the GP method, and we have various level of feature description. If we use the shorter length of symbols to describe the feature, we have a simple form to detect the time series based on the overview. On the other hand, if we used longer length of symbols, then we can recognize the time series in more details.

2.2 Feature description of time series by using primitives

In the paper, we assume that the time series is approximated by some arithmetic operations to the basic patters (called as primitives in the following) as depicted in Figure 1. These arithmetic expressions are represented as tree structures as in ordinary arithmetic expressions. Then, we call them the individuals similar to the representation in the pool of individuals of GP procedures.

If we generate these representation for a group of time series which are assumed to belong the same cluster K, then we can use them as the pool of individuals for clustering unknown time series whether the time series is classified to K or not.

The number of primitives in the paper is 9, and are stored as given data in the system. It is possible to define various operations for the primitives, but to keep the robustness of the GP procedure to approximate the time series, we restrict ourselves to only two types of operations, namely, the addition (concatenation) and multiplication. These operations are represented by $+$ and $\times$ as in ordinary arithmetic expressions.

Given two primitives denoted as a and b. Then, the addition $+$ means the simple concatenation a with b in the order while the ending of a and the beginning of b are adjusted to the same value. It may possible that a or b are not the simple primitive but the intermediate result of operations done beforehand. But, we apply the same concatenation a with b by adjusting the level of two segments of time series. The result of addition operation is stored as the intermediate segment by normalizing the time series between 0.1 and 1.0.

In the multiplication operation $\times$, at first the length of two primitives (or segments of times series) are adjusted to the same length by using the interpolation. Then, we multiply the values of corresponding time points in a and b. In a similar manner, the result of multiplication is stored as the intermediate segment by normalizing the time series between 0.1 and 1.0.

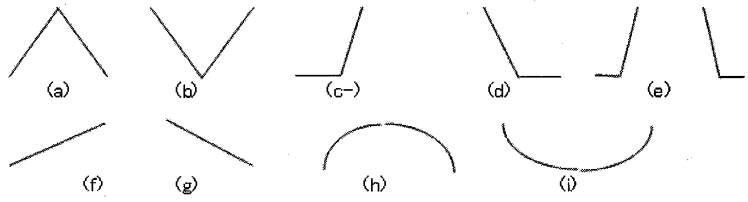

Figure 1: Overview of primitives

2.3 Overview of systems

In the following, we show the overview of the system for syntactical recognition for time series and clustering method proposed in the paper based on the GP.

(1)Learning data

It is assumed that the time series data is stored and available in the system, each of which is divided into the same length. Moreover, it is assumed that the time series data used for learning process is available, each of which is accompanied with the cluster (category) to which the underlying time series is expected to be classified. In Figure 2 depicting the learning phase, it is assumed that such kinds of learning data are prepared for each clusters $i(i = 1 \sim n)$.

(2)Learning based on the GP

At first, the approximations of functional forms (arithmetic expressions for primitives) for each cluster are obtained to describe the generating models (features) of time series. These functional forms correspond to the individuals in pool in the GP procedure. The GP method is used for the approximation of functional forms. The functional form (individual) is not a single form, but is composed of a set of forms to approximate various generating models, while learning data includes variation of time series obtained from a single basic form by expanding or shrinking the time scale. Then, in the pool used for the GP method a number of individuals having relatively higher ability (called fitness) of approximation are retained in the system, and are used for clustering.

Since we use a set of learning data repeatedly to improve the fitness of individuals, the maximum value of the fitness of individuals in the pool does not increase monotonically, which is different form ordinary optimization processes or approximation using the GP.

Therefore, the iteration of the GP procedure is carried out until sufficient variation of functional form (individuals) for approximating the feature of time series is obtained.

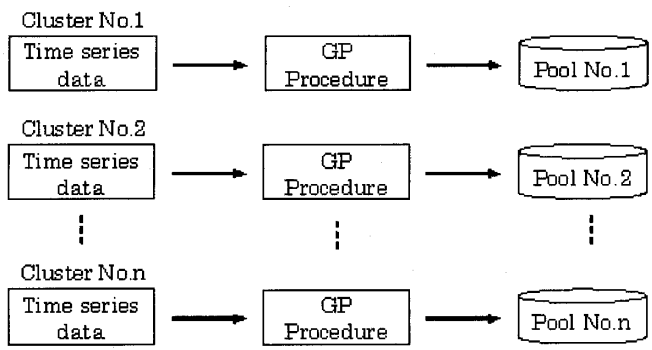

Figure 2: Overview of systems(Learning Phase)

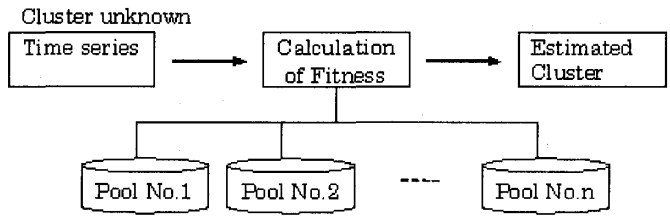

Figure 3: Overview of systems(Clustering Phase)

As is shown in Figure 2, in the Learning Phase the independent pools of individuals are organized for each cluster $i(i = 1 \sim n)$.

(3)Calculation of fitness of individuals

After applying the Learning Phase, we calculate the fitness of individuals in the Clustering Phase in Figure 3 for every individual stored in each pool $i(i = 1 \sim n)$ by adopting (fitting) the observed data $x(t)$ of underlying time series whose cluster is not known and is needed to be determined by the system. As mentioned previously, the pool P_i for cluster i includes a number of Individuals (say 1000) which are represented in syntactical expressions. We denote them as S_{ij} (jth individual in ith pool).

Then, we estimate the syntactical expression for the time series $x(t)$ by applying similar procedure depicted in Learning Phase using the GP. We obtain a syntactical form R for the time series $x(t)$. Then, we calculate the difference D_{ij} between R and each S_{ij}. The fitness f_{ij} for S_{ij} is defined as the inverse of D_{ij}.

Since the individuals S_{ij} and the expression R can be interpreted as binary tree structures where intermediate nodes correspond to operators $+$ and $\times$, and the terminal nodes corresponds to primitives. Then, we calculate the difference between two tree structure for S_{ij} and R. The procedure is defined as follows.

(i) levels of binary tree and weight

We call the root node of the tree as the low (shallow) level (level 0) in tree structure. The terminal nodes are usually in deepest levels. We also assign the weight to the level k denoted as $w_k(0 < w_k < 1)$.

(ii) compare two trees

Then, we simply compare two trees correspond to R and S_{ij} from level 0 to the largest level k. Then, we assign the score Y to this comparison. If the nodes of two trees placed on the same position are the same, then we increase Y at an amount I, otherwise we decrease Y at an amount D . These I and D are multiplied by the weights w_k corresponding to the level k of trees.

(iii) estimate cluster

Then, we estimate (determine) the cluster K of the time series $x(t)$ by selecting the highest f_{max} among f_{ij} where the individual i belongs to the K th pool.

The process is different from ordinary calculation of fitness of individuals which are represented in arithmetic expressions (functional forms)[23][24]. In these cases, we calculate the fitness of individuals by fitting nonlinear prediction functions realized by the individuals. Since the individual represented in the functional form is regarded as the prediction of time series at time t using data up to time $t-1$, we can calculate the prediction of $x(t)$ denoted as $\hat{x}(t)$. If the root mean square error (called $rmse$) between $x(t)$ and $\hat{x}(t)$ is small, then the fitness (ability) of a certain individual i realizing the underlying functional form is relatively high. Then, the fitness f_i of individual i is defined by the inverse of $rmse$.

2.4 Effectiveness of Learning Classifier Systems

In the ordinary process of optimization and functional approximation using the GP, the individual having highest fitness is selected as the best solution[9]-[15]. However, as we explained in the overview of the system, in the paper we utilize a set of individuals expected to provide us relatively higher fitness are also retained in the system, while they may be useful in the next stage of clustering for another time series.

It is shown in various studies, in learning system based on the genetic operations, it is found in the fluctuating environment, the selective utilization of individual having highest fitness leads us sometime to a wrong and unstable decision. Then, to avoid the unstability, so-called Learning Classifier System (LCS) is approved in which a part of the individuals having relatively higher fitness is used for decision making, and an individual is selected from such group [21]. Then, we can avoid sudden loss and unstable decision, for example in investments, by mitigating the decision error.

In the same way, the model of the time series belonging to the same cluster is not able to be described by a single functional form, and we must prepare a set of variation of functional form for time series which is generated by expanding or shrinking the time scale of a basic time series. The LCS can confirm the possibility to avoid the unstable clustering process.

3 Applying the GP to obtain syntactical expressions

3.1 Representation of syntactical expressions

We consider the problem to obtain a syntactical expression using primitives which approximate the time series denoted as $x(t)$. Suppose that some of $x(t)$ is actually observed, and then we estimate the arithmetic expression using primitives and two operations $+$ and $\times$ among them.

The GP is an extension of the conventional GA in which each individual in the population (pool of individuals) is a computer program composed of the arithmetic operations, standard mathematical expressions and variables [9]-[20].

In the GP, the system equations are represented in the tree structure (called individuals). The prefix representation is approved in [20] based upon the comparative study with other representation such as pointer based implementation and the postfix approach. In the parse tree the node non-terminal are taken from binomial operations $+, \times$. Terminal nodes consist of arguments chosen from set of primitives.

The prefix representation follows traditional representation by using the Lisp syntax. For example, we have the next prefix representation.

$$x(t) = (a+b) \times (c+d) \rightarrow \times + ab + cd \tag{2}$$

The equation represented by using the prefix are interpreted based upon the stack operation. We begin to scan the prefix representation, and if we meet the terminal (operand) then we push down the term into the stack by storing the result into the stack again.

For checking the validity of underlying parse tree, the so-called stack count (denoted as *StackCount* in the paper) is useful [9]-[18][22] . The *StackCount* is the number of arguments it places on minus the number of arguments it takes off from the stack. The cumulative *StackCount* never becomes positive until we reach the end at which point the overall sum still needs to be 1.

By using the *StackCount* we can see which loci on the prefix expression is available to cut off the tree for the crossover operation, and we can validate whether the mutation operation is

allowed.

If final count is 1, then the prefix representation (tree) corresponds properly to a system equation. Otherwise, the tree structure is not relevant to represent the equation.

Usually, we calculate the root mean square error ($rmse$) between $x(t)$ and $\tilde{x}(t)$ where $\tilde{x}(t)$ is the prediction of $x(t)$, and use it as the fitness. By selecting a pair of individuals having higher fitness, the crossover operation is applied to generate new individuals.

3.2 Algorithm of the GP

By using the measure of fitness to evaluate each individual, we apply the GP to the population to derive better description for the dynamics which generates the targeted time series. Figure 2 shows the overview of the Genetic Algorithm.

Crossover operations

Contrary to the operation in GA, the crossover operation in GP is applied to restricted cases. Then, we can not choose arbitrary loci in the string of individuals and replace the parts of two tree structures. The two subtrees are extracted and swapped each other.

To keep the crossover operation always producing syntactically and semantically valid programs, we look for the nodes which can be a subtree in the crossover operation and check for no violation. By using the *StackCount* already mentioned, we know the subtrees which are the candidate for the crossover operation. The basic rule is that any two loci on the two parents genomes can serve as crossover points as long as the ongoing *StackCount* just before those points is the same. The crossover operation creates new offsprings by exchanging sub-trees between two parents.

Mutation

The goal of the mutation operation is the reintroduction of some diversity in an population. Two types of mutation operation in GP is utilized to replace a part of the tree by another element.

(Global mutation :G-mutation)

Generate a individual I_s, and select a subtree which satisfies the consistency of prefix representation. Then, select at random a terminal in the individual, and replace the terminal by the subtree of the individual I_s.

(Local mutation:L-mutation)

Select at random a locus in a parse tree to which the mutation is applied, we replace the place by another value (a primitive function or a variable).

We iteratively perform the following steps until the termination criterion has been satisfied.

(Step 1)

Generate an initial population of random composition of possible functions and terminals for the problem at hand. The random tree must be syntactically correct program.

(Step 2)

Execute each individual (evaluation of system equation) in population by applying the optimization of the constants included in the individual. Then, assign it a fitness value giving partial credit for getting close to the correct output. Then, sort the individuals according to the fitness S_i.

(Step 3)

Select a pair of individuals chosen with a probability p_i based on the fitness. The probability p_i is defined for ith individual as follows.

$$p_i = (S_i - S_{min}) / \sum^{N} (S_i - S_{min}) \tag{10}$$

where S_{min} is the minimum value of S_i, and N is the population size.
(Step 4)

Then, create new individuals (offsprings) from the selected pair by genetically recombining randomly chosen parts of two existing individuals using the crossover operation applied at a randomly chosen crossover point. Iterate the procedure several times to replace individuals with lower fitness.
(Step 5)

If the result designation is obtained by the GP (the maximum value of the fitness become larger than the prescribed value), then terminate the algorithm, otherwise go to Step 2.

4 Application to Clustering of time series

4.1 Approximation for artificially generated time series

To test the capability of clustering method of the paper, we apply the system to the time series which are artificially generated by transforming the time scale for the original time series. At the beginning, we prepare 10 observations of time series, and they are regarded to be generated by using independent known generating function. For simplicity, we generate 10 independent time series $x_i(t), i = 1 \sim 10$ by using prefix representation for the arithmetic expression using primitives generated at random. But, we remove very resemble time series by observing the result of time series generations. We call these prefix representation for each time series $x_i(t)$ as the $A_i, i = 1 \sim 10$.

Then, we apply the GP operations to approximate these time series $x_i(t)$ to obtain prefix representations, and we call them as the B_i corresponding to A_i. We denote the time series approximated by B_i as $y_i(t)$. Theoretically, the time series generated by two processes $x_i(t)$ and $y_i(t)$ are very close, and two representations A_i and B_i are very resemble. However, the approximation may be depend on the number of GP generations.

Then, we show the result for approximation depending on the GP generations. The conditions for the simulation study are summarized as follows.
maximum length of array in individuals:20, 30 and 40
number of individuals in each pool:1000

The result of simulation is shown in Table 1. In the table, l is the maximum length of array in individuals. The average of the root mean square error between $x_i(t)$ and $y_i(t)$ (denoted as $rmse$) are depicted along with the GP generations (denoted as GP-g). As is shown in Table 1, the approximation of the time series using the primitives are almost obtained in 100 GP generations.

Table 1-Result of approximation

l	GP-g $rmse$	GP-g $rmse$	GP-g $rmse$
20	10 0.09	50 0.002	100 0.0003
30	10 0.12	50 0.08	100 0.005
40	10 0.30	50 0.10	100 0.010

4.2 Clustering for artificially generated clusters

At the next step, we apply the transformation to these original 10 time series by changing the time scale by expanding and shrinking the time scale. The function for the transformation

is called as the warping function, and is depicted in Figure 4. By applying various warping function to the original 10 time series, we have a set of clusters of time series where a group of time series generated from a certain original time series in previous section by applying warping function should belong to the same cluster.

definition of warping function

The warping function is defined as a curve obtained by overlaying a half cycle sinusoidal wave or a full cycle sinusoidal wave having the amplitude a on the straight line with 45 degree angle between two axis. Namely, the beginning and the ending points of the time scale for both original and transformed time series are the same, but intermediate sample points are shifted according to the transformation depicted in Figure 3. The sample points at time t_1 in the original time series is shifted to time t_2 in the transformed time series. The form of the warping functions are denoted as $w-a, w-b, w-c, w-d$ from the left top through the right bottom in Figure 3.

If we take 30 different values for the amplitude a, we have 30 time series for each clusters in previous section, and these 30 time series should be theoretically classified into the same cluster.

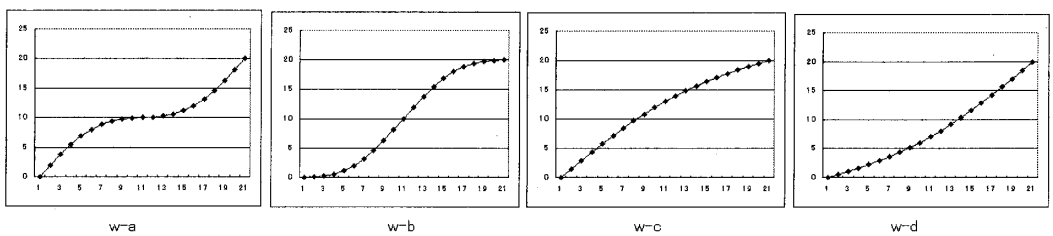

Figure 4: Definition of warping functions

Learning by the GP

Then, we apply the Learning Phase to the generated times series for each cluster. The condition for the learning by the GP is summarized as follows.

Maximum length of array of individual:20

Number of individuals in pool:1000

Maximum generation of GP:900

The way of learning is given as follows. Generated 30 time series are repeatedly used for learning, namely, a time series is used for 30 generation and the time series for learning is switched to another time series. A set of 1000 individuals is assigned as a pool to a cluster among 10 clusters, and the fitness of the individuals is improved by applying the GP procedure at 900 times. Finally we obtain a LCS composed of individuals possessing higher fitness for clustering the time series.

By the way, the maximum number of the GP generation is limited to 900, while our purpose is to obtain a pool of individual having relatively higher fitness as a group rather than the optimization or the ultimate approximation of the problem. Since 30 time series are repeatedly used for learning for the Learning Phase including the original time series as well as transformed time series, the maximum fitness of the pool of the individuals dose not increase monotonically.

simulation result

Table 2 summarizes the result of classification for the time series in No.1 cluster by using the system proposed in the paper. The value in Table 2 shows the rate of time series which are truly classified to cluster No.1 among the total 30 time series. For simplicity, the data for learning and testing are the same in the simulation study. In Table 2, the result for four

warping function are summarized. As a result, the ability of the system for clustering artificially generated time series is good enough.

Table 2-Result of clustering(probability of true clustering %)

w-a	w-b	w-c	w-d
98.0	98.0	97.0	98.0

5 Application to clustering of segments in stock prices

5.1 Segments in stock prices

Now we apply the method of the paper to clustering of segments included in the stock trends (prices). In the technical analysis of stock price, the parts of the stock price (called as segments) are characterized with their features as almost standardized forms [21]. Figure 6 shows these basic 8 patterns of the segments as rough sketches. It is known that traders forecast the rise/fall of underlying stock by checking and recognizing the appearance of these 8 patterns of segments The notations and characteristics of these 8 segments are summarized as follows.
(a)down trends
prices fall monotonically with small fluctuations.
(b)uptrends
prices rise monotonically with small fluctuations.
(c)head and shoulder
relatively high one peak like a head is located between two relatively lower peaks like shoulders.
(d)broadening
prices move inside a triangular envelope increasing its amplitude.
(e)breakout
prices suddenly rise and then increase monotonically.
(f)rectangular
prices move inside a envelope composed of two parallel lines.
(g)rounding
prices move along a large monotonic curve showing their peaks on the ends.
(h)triangle prices move inside a triangular envelope decreasing its amplitude, and finally converge to constants.

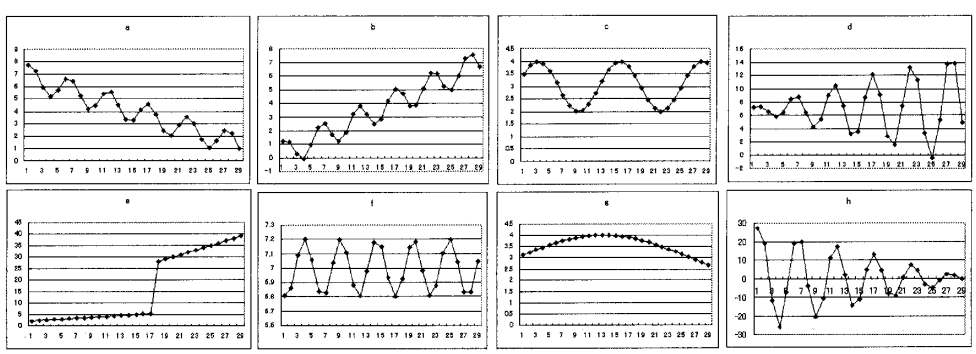

Figure 5: Overview of 8 patterns of segments in stock prices

Clustering segments of stock price

Since the process of clustering of the time series described in the preceding sections is the same for the clusters of segments of stock price, we concisely summarize only the result of simulation for clustering of segments. The condition for the simulation study is as follows:
Number of time series for each cluster:30
Maximum length of individual:20
Number of individuals in pool:1000
Maximum generation of GP:600

For simplicity, the data for learning and testing are the same in the simulation study.

Table 3 shows the result of clustering. In Table 3, the numbers in vertical column mean the original clusters to which the time series belong, and the numbers in horizontal row mean the estimated clusters obtained by the system which is determined by clustering process. The values in the table denote the rate of clustering. Theoretically, only the diagonal elements take the value of 100 %, but due to the classification error, a part of time series are misclassified into another clusters.

However, the rate of proper clustering ranges from 70 % through 97%, and the average value of proper clustering is about 85 %. Then we can confirm the ability of clustering system.

Table 3. Result of clustering of segments(%)

No.	a	b	c	d	e	f	g	h
a	80.0			10.0				
b		83.3		6.7			3.0	
c			93.3	6.7				
d		13.3		76.7			3.0	
e		6.7			93.3			
f		3.3				93.3		1.3
g		26.6					73.3	
h		20.0		10.0				70.0

5.2 Discussion

The values in Table 3 mean the average rate of true clustering for each cluster. Namely, the false cases for clustering may happen for the misclassification, the wrong detection of segments and the lost of necessary recognition of segments.

If we examine the result of error included in clustering in Table 3, relatively high rate of misclassification happen among segments having resemble trends as a whole pattern. For example, some of samples in segment d are misclassified into segment b, but two patterns b and d are resemble except for the fluctuation of prices. The upper half of envelope is the same in two patterns, but d has also lower edge of envelops. Then, it is imagined that the system overlook the lower envelope in several cases. The same misclassification is observed in segment a in which some of samples are classified into d and h

Contrast to these segments including envelopes, samples in segments c, e, f are almost truly classified into original clusters. The result reflects the basic feature embedded in segments.

6 Conclusion

This paper showed clustering of segments of stock prices by using syntactical recognition system for time series based on the GP. The pool of individuals improved by the GP for clustering a time series belongs to a certain cluster is useful to absorb the variation of the time series in

clusters. The system was applied to clustering of artificially generated time series, and also to clustering of 8 kinds of segments of real stock prices.

The problem to be solved is still remained in the configuration of upper level of recognition system for longer time series in a syntactical manner, and the future works will be done by the authors.

References

[1] A.A.Freitas, Data Mining and Knowledge Discovery with Evolutionary Algorithms, Springer-Verlag, 2002.

[2] K.S.Fu,Syntactical Method in Pattern Recognition,Academic Press, 1974.

[3] W.H.Tsai and K.S.Fu, " Attributed grammar- a tool for combining syntactical and statistical approaches to pattern recognition", IEEE Transaction on Systems, Man and Cybernetics, vol.SMC-10,no.12,pp.873-885 ,1980.

[4] P.Trahanias and E.Skordalakis,"Syntactic pattern recognition of the ECG",IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.12,no.7,pp.648-657,1991.

[5] S.Tokinaga, " On the management of time-series data by feature description based upon generative grammar- By fitting ARMA model to EEG signals", IEICE Trans.Information(Japanese Edition), vol. J67-D,vol.10,pp.1099-1106,1984.

[6] S.Tokinaga,"Automatic EEG classification based on syntactical pattern recognition method-feature extraction by adaptive ARMA model fitting",IEICE,Trans.Fundamentals, vol.E69,no.10, pp.1125-1132,1986.

[7] N.Takagi,S.Tokinaga and Y.Ikeda, " Linguistic expression of time series based on the Genetic Programming and its applications", Proc. of ISPACS 2003, pp.690-695, 2003.

[8] S.Tokinaga,J.Lu and Y.Ikeda, " Realization of syntactic recognition systems of time series based on the Genetic Programming and its applications", Proc.of 2004 DSP Symposium,2004.

[9] Y.Ikeda and S.Tokinaga, " Approximation of chaotic dynamics by using smaller number of data based upon the genetic programming, IEICE Trans.Fundamentals, vol.E83-A,no.8, pp.1599-1607,Aug.2000.

[10] Y.Ikeda and S.Tokinaga, " Controlling the chaotic dynamics by using approximated system equations obtained by the genetic programming, IEICE Trans.Fundamentals,vol.E84-A, no.9, pp.2118-2127,Sept.2001.

[11] Y.Ikeda, " Estimation of the chaotic ordinary differential equations by co-evolutionary genetic programming, Trans. IEICE Trans.Fundamentals, vol.E85-A,no.4,pp.424-433,April,2002.

[12] M.Yababe and S.Tokinaga, " Applying the genetic Programming to modeling of diffusion processes by using the CNN and its applications to the synchronization", IEICE Trans. Fundamentals(Japanese Edition),vol.J85-A,no.5,pp.548-559,May 2002.

[13] X.Chen and S.Tokinaga, " Approximation of chaotic dynamics for input pricing at service facilities based on the GP and the control of chaos , IEICE Trans.Fundamentals, vol.E85-A,no.9,pp.2107-2117,Sept.2002

[14] X.Chen and S.Tokinaga, " Synthesis of multi-agent systems based on the co-evolutionary genetic Programming and its applications to the analysis of artificial markets", IEICE Trans.Fundamentals(Japanese Edition) ,vol.E86-A,no.10,pp.1038-1048, Oct.2003.

[15] S.Tokinaga and M.Tominaga, " An improvement method for the workflow management systems based on the reallocation of flows using the Genetic Programming and its applications, Journal of Operations Research Society of Japan, vol.46, no.3, pp.286-305,2003.

[16] Y.Ikeda and S.Tokinaga, " Chaoticity and fractality analysis of an artificial stock market by the multi-agent systems based on the co-evolutionary Genetic Programming, IEICE Trans.Fundamentals ,vol.E87-A, no.9,pp.2387-2394,Sept.2004.

[17] J.R.Koza:Genetic Programming,MIT Press, 1992

[18] J.R.Koza, " Genetic programming:A paradigm for genetically breeding populations of computer programs to solve problems", Report No.STAN-CS-90-1314, Dept.of.Computer Science Stanford University, 1990.

[19] J.Koza,Evalution and subsumption using genetic programming, Proc of the First European Conference on Artificial Life, MIT Press,1991.

[20] M.J.Keith and M.C.Martin, " Genetic programming in C++: Implementation issues", in (ed) K.E.Kinnerar,Jr.,Advance in Genetic Programming MIT Press,1994.

[21] L.B.Brooker,D.E.Goldberg and J.H.Holland,"Classifier system and genetic algorithm", Artificial Intelligence, 40, pp.235-282, 1989.

[22] S.Tokinaga and Y.Ishida, " An intelligent digital signal processing systems for stock trends based upon transient wave detection by using Gabor representation and knowledge representation of waveform, IEICE Trans.Fundamentals,vol.J78-A, vol.2,pp.169-177,Feb.,1995.

[23] J.J.Fernandez, K.A.Farry and J.B.Cheatham, "Waveform recognition using Genetic Programming: The myoelectric signal recognition problem" in (eds J.Koza) Genetic Programming 1996, pp.63-71,1996.

[24] S.Tokianga and J.Lu, " On modeling of time series based on the Genetic Programming and its application to clustering (in Japanese) ",Technical Report of the IEICE,SIP2004-61,pp.71-76,2004.

Xiaorong Chen
(Antai School of Manegement,Shanghai Jiaotong University)

Shozo Tokinaga
(Graduate School of Economics, Kyushu University)