Data Mining by Soft Computing Methods for The Coronary Heart Disease Database

Akira Hara and Takumi Ichimura

Graduate School of Information Sciences, Hiroshima City University

3-4-1, Ozuka-higashi, Asaminami-ku, Hiroshima, 731-3194 Japan

email:{ahara, ichimura}@hiroshima-cu.ac.jp

Abstract—**For improvement of data mining technology, the advantages and disadvantages on respective data mining methods should be discussed by comparison under the same condition. For this purpose, the Coronary Heart Disease database (CHD_DB) was developed in 2004, and the data mining competition was held in the International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES). In the competition, two methods based on soft computing were presented. In this paper, we report the overview of the CHD_DB and the soft computing methods, and discuss the features of respective methods by comparison of the experimental results.**

I. INTRODUCTION

Prognosis of disease plays an important role in medicine. In order to enhance the accuracy of prognosis, the variety of prognostic systems have been developed using techniques that originate both from the fields of statistics and artificial intelligence. It is unlikely that one technique can always outperform the others. The advantages and disadvantages of using different techniques should be discussed to improve the performance of prognostic system.

The developed prognostic system needs to be compared with existing systems under the same procedure and the same database. However, few medical databases are made available to researchers, because it is impossible to distribute medical data without ensuring the protection of privacy and the confidentiality of information. In order to resolve this impasse, Suka *et al.* have been developed the Coronary Heart Disease Database (CHD_DB) that consists of four training datasets and one testing dataset[1]. Any authorized researcher can construct a prognostic system using the training datasets and measure its performance using the testing dataset.

The data mining competition using the CHD_DB was held in the International Conference on Knowledge-Based Intelligent Information and Engineering Systems. In the competition, rule extraction method using Automatically Defined Groups (ADG)[2] and classification method using Immune Multiagent Neural Network (IMANN)[3] were presented as soft computing methods. ADG is a method based on Genetic Programming[4], and IMANN based on neural network and immune systems. In this paper, we report the overview of the CHD_DB and these methods. In addition, we compare the experimental results, and discuss the advantages and disadvantages on respective methods.

The contents of this paper are as follows. Section II describes the detail of CHD_DB. Section III describes the pre-viously developed soft computing methods, and experimental results. Especially, Section III-A explains rule extraction by ADG, and Section III-B explains classification by IMANN. Section IV describes the conclusions.

II. CORONARY HEART DISEASE DATABASE[1]

The CHD_DB is designed to reproduce the original data of the Framingham Heart Study. Requisite information is derived from the reports on six-year follow-up in the Framingham Heart Study[5].

Table I shows the data items of the CHD_DB. Each of the datasets consists of ten data items: ID, development of CHD, and eight items that were collected from the initial examination. The eight items were examined whether it was associated with the development of CHD over six-year follow-up in the Framingham Heart Study. Using the CHD_DB, researchers can develop a prognostic system that will discriminate between those who developed CHD (CHD cases) and those who did not (Non-CHD cases) on the basis of the eight data items.

The CHD_DB consists of four training datasets (Train_A, X, Y, and Z) and one testing dataset. The four training datasets are designed to have different proportions of CHD cases to Non-CHD cases. In the original data of the Framingham Heart Study, the number of complete records was about four thousand and the proportion of CHD cases to Non-CHD cases was about one to nine (*i.e.* the six-year incidence of CHD was 9.1%). In CHD_DB, Train_A, Train_X and Train_Y include 6,500 CHD cases and 6,500 ($\times 1$), 13,000 ($\times 2$) and 585,000 ($\times 9$) Non-CHD cases, respectively, while Train_Z includes 400 CHD cases and 3,600 Non-CHD cases. The Train_Z approximates to the original data both for the number of total records and the proportion of CHD cases to Non-CHD cases. On the other hand, the testing dataset includes 6,500 CHD cases and 6,500 Non-CHD cases.

Validity check of the four training datasets and the one testing dataset had been performed using statistical analysis methods. First, percent distribution of the categorical data items and means and standard deviations of the numerical data items were equal to those of the original data. Second, to examine the association between the eight data items and the development of CHD, odds ratios and the 95% confidence intervals for the development of CHD were calculated using multiple logistic regression models. A significantly high or low odds ratio was found in education, tobacco, left ventricular

TABLE I

Data Item	Name	Value
ID	ID	Sequential Value
Development of CHD	CHD	0=No; 1=Yes
National Origin	ORIGIN	0=Native-born; 1=Foreign-born
Education	EDUCATE	0=Grade School; 1=High School, not graduate; 2=High School, graduate; 3=College
Tobacco	TOBACCO	0=Never; 1=Stopped; 2=Cigars or Pipes; 3=Cigarettes (<20/day); 4=Cigarettes ($\geq$20/day)
Alcohol	ALCOHOL	Continuous Value (oz/mo)
Systolic Blood Pressure	SBP	Continuous Value (mmHg)
Diastolic Blood Pressure	DBP	Continuous Value (mmHg)
Cholesterol	TC	Continuous Value (mg/dl)
Left Ventricular Hypertrophy	LVH	0=None; 1=Definite or Possible

hypertrophy, systolic blood pressure, diastolic blood pressure, and cholesterol, but not in national origin and alcohol. The results from the four training datasets and the one testing dataset were in good agreement with the result from the original data.

The developed prognostic system will be evaluated in terms of (1) classification accuracy for total, positive, and negative cases, (2) area under receiver operating characteristic curve, and (3) concordance index, which indicate the discrimination ability (*i.e.* how well the prognostic system discriminates between CHD cases and Non-CHD cases). The use of the CHD_DB will enables the researchers to discuss the advantage and disadvantage of different techniques. Such discussion will contribute to improve the performance of prognostic system and data mining methods.

III. DATA MINING BY SOFT COMPUTING METHODS

A. Rule Extraction Using Automatically Defined Groups[2]

In [2], rule extraction method using Automatically Defined Groups (ADG), which is based on Genetic Programming[4], had been proposed, and the performance was examined by using the CHD_DB. The aim of the method is to extract not only general diagnostic rules but also exceptional diagnostic ones. The another aim is not only to improve the classification accuracy but also to acquire the useful and comprehensible knowledge.

In the method, a multi-agent approach is used in order to perform both the clustering of data and rule extraction from each cluster. That is, the data are divided among agents. This corresponds to the clustering of data. And each agent generates a rule for the assigned data. This corresponds to the rule extraction in each cluster. As a result, multiple rules are extracted by multi-agent cooperation.

ADG is a method to optimize both the grouping of agents and the program of each group in the process of evolution. By grouping multiple agents, we can prevent the increase of search space and perform an efficient optimization. Moreover, we can easily analyze the agents' behavior. The acquired group structure is utilized for understanding how many roles are needed and which agents have the same role.

A team that consists of all agents is regarded as one GP individual. One GP individual maintains multiple trees, each of
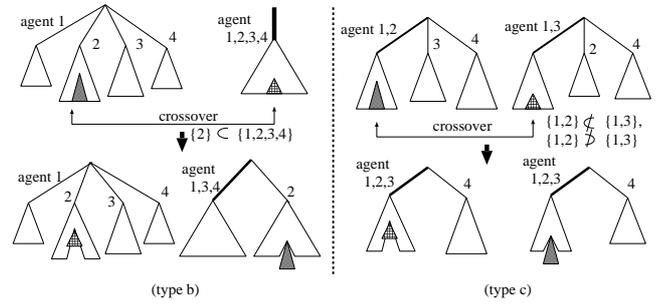

Fig. 1.   Examples of crossover in ADG.

which functions as a specialized program for a distinct group. We define a group as the set of agents referring to the same tree. All agents belonging to the same group use the same program.

Generating an initial population, agents in each GP individual are divided into random groups. Basically, crossover operations are restricted to corresponding tree pairs. For example, a tree referred to by an agent 1 in an individual breeds with a tree referred to by an agent 1 in another individual. In addition, we also consider the sets of agents that refer to the trees used for the crossover. The group structure is optimized by dividing or unifying the groups according to the relation of the sets. Individuals search solutions as their group structures gradually approach the optimal structure.

The concrete processes are as follows: We arbitrarily choose an agent to two parental individuals. A tree referred to by the agent in each individual is used for crossover. T and T' are used as expressions of these trees, respectively. In each parental individual, we decide a set $A(T)$, the set of agents that refer to the selected tree T. When we perform a crossover operation on trees T and T', there are the following three cases.

(type a) If the relation of the sets is $A(T) = A(T')$, the structure of each individual is unchanged.

(type b) If the relation of the sets is $A(T) \supset A(T')$, the division of groups takes place in the individual with T, so that the only tree referred to by the agents in $A(T) \cap A(T')$ can be used for crossover. The individual which maintains T' is unchanged. Fig. 1
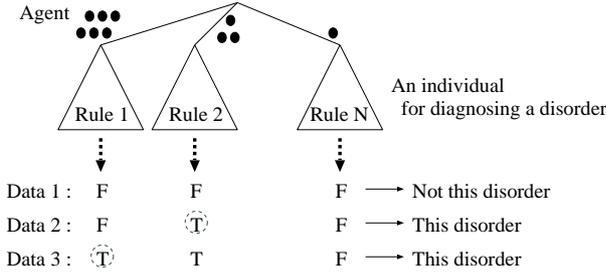
Fig. 2. Diagnostic system for a particular disorder.

shows an example of this crossover.

(type c) If the relation of the sets is $A(T) \not\supset A(T')$ and $A(T) \not\subset A(T')$, the unification of groups takes place in both individuals so that the agents in $A(T) \cup A(T')$ can refer to an identical tree. Fig.1 shows an example of this crossover.

By using this method, the search works efficiently and the adequate group structure is acquired. Besides, the acquired group structure becomes a clue for understanding the cooperative behavior and necessary division of labor.

1) How to Apply ADG to CHD_DB: In order to judge whether each data is regarded as CHD case, we will find logical expressions such that only the data in CHD cases should satisfy. The logical expression is made by the conjunction of multiple terms. Each term is the combination of a test item and the normalized value which can be taken. The logical expression has to return false for Non-CHD cases. The following expression is an example.

Rule for CHD : $(TC > 0.51) \wedge (TC < 0.68) \wedge (DBP > 0.49)$

In medical field, the diagnoses are largely dependent on each doctor's experience. Therefore, the diagnostic rule is not necessarily represented by a single rule. Moreover, some data can be classified into different results, even if the results of the tests are the same. We apply ADG to the diagnoses of coronary heart diseases with consideration of this background.

We describe the detail of rule extraction for CHD cases. Multiple trees in an individual of ADG represent the respective logical expressions. Each data in the training set is input to all trees in the individual. Then, calculations are performed to determine whether the data satisfy each logical expression. As illustrated by data 2 in Fig.2, the input data is regarded as CHD case if one or more logical expressions in the individual return true. In contrast, as illustrated by data 1 in Fig.2, the input data is not regarded as CHD case if all logical expressions in the individual return false.

The concept of each agent's load arises from the viewpoint of cooperative problem solving by multiple agents. The load is calculated from the adopted frequency of each group's rule and the number of agents in each group. The adopted frequency of each rule is counted when the rule successfully returns true for each CHD data. As illustrated by data 3 in Fig.2, if multiple trees return true for a CHD data, the tree with more agents is

adopted. When the k-th agent belongs to group g, the load of the agent is defined as follows.

$$w_k = \frac{(\text{adopted frequency of } g) \times N_{agent}}{(\text{Number of agents which belong to } g) \times N_{all\_adoption}}$$

In this equation, N_{agent} represents the number of all agents in one GP individual, and $N_{all\_adoption}$ represents the sum of adopted frequencies of all groups. By balancing every agent's load, more agents are allotted to the group that has a greater frequency of adoption. On the other hand, the number of agents in the less adopted group becomes small. Therefore, we can acquire important knowledge about the ratio of use of each rule. The ratio indicates how general each rule is for judgment of the disorder. Moreover, when other cases are judged to be true through a mistake of a rule, it is thought that the number of agents who support the rule should be small.

To satisfy the requirements mentioned above, fitness f is calculated by the following equation. We maximize f by evolution.

$$\begin{aligned} f = & -\frac{miss\_target\_data}{N_{CHD}} - \alpha \frac{misrecognition}{N_{NonCHD}} \\ & -\beta \frac{\sum_{N_{NonCHD}} fault\_agent}{misrecognition \times N_{agent}} - \delta V_w \end{aligned} \quad (1)$$

In this equation, N_{CHD} and N_{NonCHD} represent the number of CHD cases and Non-CHD cases in database respectively. $miss\_target\_data$ is the number of missing data in the target CHD data that should have been judged to be true. $misrecognition$ is the number of mistakes through which Non-CHD data is regarded as CHD case. When the rule returns true for Non-CHD data, $fault\_agent$ is the number of agents who support the wrong rule in each data. So, the third term represents the average rate of agents who support the wrong rules when misrecognition happens. V_w is the variance of every agent's load. In addition, in order to inhibit the redundant division of groups, f is multiplied by γ^{G-1} ($\gamma > 1$) according to the increase of the number of groups, G, in the individual.

By evolution, one of the multiple trees learns to return true for a data in the CHD cases, and all trees learn to return false for Non-CHD cases. Moreover, agents are allotted to respective rules according to the adopted frequency, and the allotment to a rule with more misrecognition is restrained. Therefore, the rule with more agents is the typical and reliable diagnostic rule, and the rule with less agents is an exceptional rule for the rare case.

The following points are regarded as the advantages of ADG.

- ADG enables us to extract rules for exceptional data that is likely to be missed by a single rule.
- It is easy to judge by the number of agents whether the acquired rules are typical ones or exceptional ones.
- It is easy to understand the acquired rules, because typical rules and exceptional rules are clearly separated.

Table II shows GP functional and terminal symbols. The parameter settings of ADG are as follows: Population size is 500, and the number of agents is 50.

Symbol	#args	functions
and	2	arg0 ∧ arg1
gt	2	if (arg0 > arg1) return T else return F
lt	2	if (arg0 < arg1) return T else return F
TC, SBP, ...	0	normalized test value
0.0 − 1.0	0	real value

TABLE III
CLASSIFICATION ACCURACY.

Dataset	Training	Test
Train_A	70.0%	67.8%
Train_X	70.2%	68.5%
Train_Y	70.1%	68.6%
Train_Z	75.0%	66.6%

Fig. 3.   Change of the average of the number of groups.

2) Experimental Results for CHD_DB: In this section, ADG is applied to the training data so that only CHD cases can satisfy the rules. We describe the detail of an experiment using Train_Z. The respective weights in equation(1) are $\alpha = 1.0$, $\beta = 0.0001$, $\delta = 0.01$, and $\gamma = 1.001$.

Fig. 3 shows the average group number by generation. The number of groups corresponds to the number of extracted rules. As a result, 50 agents in the best individual were divided into 12 groups. We show the acquired rules that correspond to the tree structural programs in the best individual. Rules are arranged according to the number of agents that support each rule, and each terminal real value is transformed to original range. The rules with more agents are frequently adopted rules. The rules with less agents are rules for exceptional data.

Rule 1 (19 Agents): (SBP > 179)

Rule 2 (7 Agents): (LVH = 1)

Rule 3 (6 Agents): (TC > 199) ∧ (SBP > 141) ∧ (DBP > 99) ∧ (DBP < 112) ∧ (LVH = 0) ∧ (EDUCATE < 3) ∧ (ALCOHOL < 34.54)

Rule 4 (6 Agents): (TC > 264) ∧ (SBP > 150) ∧ (TOBACCO > 1) ∧ (ALCOHOL < 44.9)

Rule 5 (2 Agents): (TC > 168) ∧ (TC < 252) ∧ (SBP > 127) ∧ (DBP > 106) ∧ (TOBACCO > 2) ∧ (ALCOHOL > 19.0)

Rule 6 (2 Agents): (TC > 310)

Rule 7 (2 Agents): (SBP > 141) ∧ (DBP > 104) ∧ (LVH = 0) ∧ (EDUCATE < 2) ∧ (TOBACCO > 0) ∧ (TOBACCO < 3)

Rule 8 (2 Agents): (TC > 242) ∧ (TC < 296) ∧ (DBP > 109) ∧ (ORIGIN = 1) ∧ (TOBACCO > 0) ∧ (ALCOHOL > 15.9)

Rule 9 (1 Agents): (TC > 214) ∧ (SBP > 152) ∧ (DBP > 85) ∧ (EDUCATE < 1) ∧ (TOBACCO < 2)

Rule 10 (1 Agents): (DBP > 79) ∧ (DBP < 84) ∧ (ALCOHOL > 37.5)

Rule 11 (1 Agents): (TC > 233) ∧ (SBP > 160) ∧ (DBP > 98) ∧ (DBP < 132) ∧ (ORIGIN = 0) ∧ (EDUCATE < 3) ∧ (ALCOHOL < 35.1)

Rule 12 (1 Agents): (TC > 186) ∧ (TC < 330) ∧ (SBP > 169) ∧ (DBP > 99) ∧ (DBP < 114) ∧ (LVH = 0) ∧ (TOBACCO > 0) ∧ (TOBACCO < 3) ∧ (ALCOHOL < 34.5)

The judgment accuracy for 4,000 training data is as follows. One or more rules return true for 308 cases of 400 CHD cases, and all ru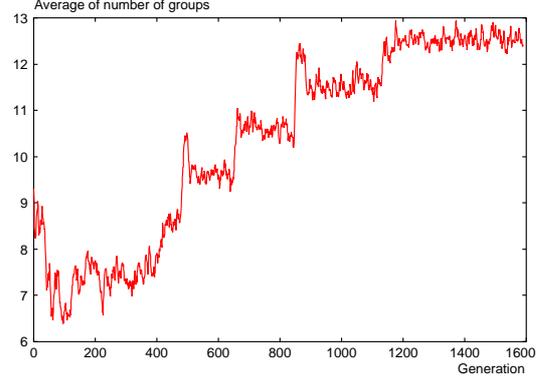les successfully return false for 2,691 of 3,600 Non-CHD cases. Therefore, the classification accuracy of the training data is 75.0%.

It was examined which rule's output is adopted for the 308 successful data. The counts of adoption of these twelve rules are 115, 46, 38, 36, 16, 13, 12, 10, 9, 7, 4, and 2 times, respectively. The rule with more agents tends to have a higher adopted frequency and more reliability. Both typical rules for frequent cases and exceptional rules for rare cases were extracted successfully. Moreover, this system was applied to 13,000 test data. As a result, it succeeded in the classification of 8,655 cases. The classification accuracy was 66.6%.

This method was also applied to other training data sets (Train_A, X, Y), and the performance for both training and test data was examined. Table III shows the classification accuracy. The acquired rules are represented by simple logical expressions. So, users can easily acquire diagnostic knowledge from the rules. However, constraint of the expressions may have a bad influence upon the classification accuracy. By modifying the GP symbols so that the rules can represent more complex expressions (*e.g.* DBP > 1.2SBP etc.), it is necessary to improve the classification accuracy with keeping the comprehensibility.

B. Classification by Immune Multi-agent Neural Network[3]

In [3], Immune Multi-agent Neural Network (IMANN) had been proposed, and the performance was examined by using the CHD_DB.

Reflective Neural Networks, proposed by Ichimura *et al.*, is a kind of module nets, which can handle subsets of the complete set of training cases. The architecture of the reflective NN is based on the network module concept. There are two kinds of network modules: an allocation module to distribute training cases and some classification modules to classify a subset of training cases. Reflective NN has an outstanding classification capability, even if there are missing values.

However, the optimal number of classification modules in the reflective NN can not be determined according to the probability distribution function in the space of training cases. To solve such a problem, Immune Multi-agent Neural Networks (IMANN) has been proposed. The method can find

the relation between the number of modules and the number of subsets of training cases. The IMANN has macrophage agents, T-cell agents and B-cell agents. The macrophage and T-cell employ the planar lattice neural networks (PLNN) with the neuron generation/annihilation algorithm. This network structure consists of hidden neurons in the lattice. The network can work in a manner similar to self-organized map (SOM). B-cell employs Darwinian neural networks (DNN), which have a structural learning algorithm based on Darwin's theory of evolution.

1) Planar Lattice Neural Network: PLNN is a type of three-layered neural network, but the neurons in the hidden layer are arranged on a lattice. The network can work like SOM, that is, the patterns between inputs and outputs are classified into some groups in the lattice.

Fig.4 shows an overview of PLNN. The network is a kind of layered neural networks, consisting of an input layer, an interconnected hidden layer, and an output layer. The interconnected hidden neurons adjust the connection weights between input neurons and hidden neurons according to the relation of input-output patterns and the neighborhood of the hidden neuron. The adjustment of the input weight vectors follows the Kohonen learning algorithm[6]. The neighboring neurons on the network tend to be similar to the input weight vectors and the weights represent neighboring regions in the input pattern space. In particular, the asymptotic values of the weight vectors will tend towards the weighted center of their influence regions. The adjustment of the weights between hidden neurons and output neurons follows the back propagation algorithm.

The neurons in the lattice are added or eliminated by the generated/annihilated algorithm according to monitoring the variance of weight vector.

2) Immune Cells by PLNNs: Macrophage employs the PLNN to classify the training cases. The hidden neurons are generated/annihilated during the learning phase, and consequently the remaining neurons are assigned to the corresponding subset of training cases, respectively. T-cell employs neural network learning to assign each training case into one of the B-cells. T-cell employs the lower part of PLNN and the network enforces learning reverse signal from output neurons as shown in Fig.5. Because T-cell also recognizes input signals, T-cell trains the lower part of PLNN simultaneously. The teaching signals in the network consist of binary strings: 1 (on) and 0 (off). In biological immune systems, B-cells receive stimulations from T-cells. In IMANN model, B-cell employs a simple DNN learning method to train a network for the subset of training cases assigned by T-cell, as shown in Fig.6. Although B-cells work to learn a subset of training cases independently, B-cells cooperate with each other in a classification task.

Fig.7 shows the reasoning process of the trained IMANN. After training PLNN, an arbitrary input is given to T-cell NN. T-cell NN classifies into a group and stimulates the corresponding B-cell NN. The B-cell NN calculates output activities as a total output of IMANN.
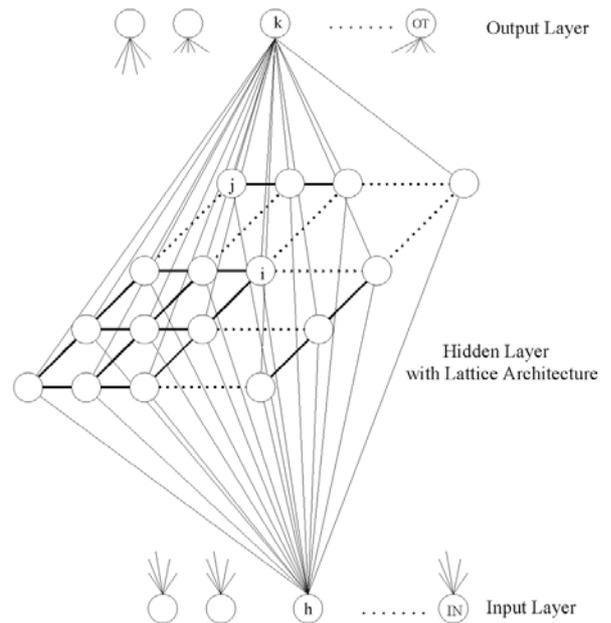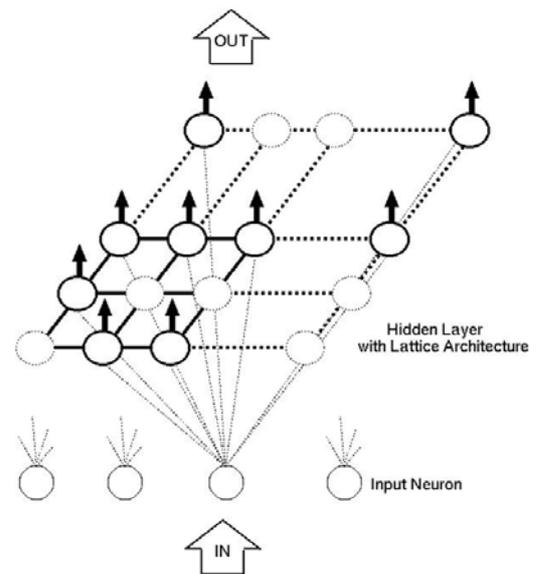

Fig. 4. Planar lattice neural networks.

Fig. 5. T-cell neural network.

3) Experimental Results for the CHD_DB: The IMANN was applied to the Train_A in CHD_DB, which are consisted of 13,000 cases. The macrophage (PLNN) has 20×20 squared neurons in hidden layer by the structure level adaptation as shown in Fig.8. 14 neurons were obtained in the lattice. T-cell learns the relation between an input pattern and its allocated categories using PLNN. B-cell trains the neural network for 14 subsets of training cases, respectively. The classification accuracy of the test dataset is 82.3% (10,699/13,000).
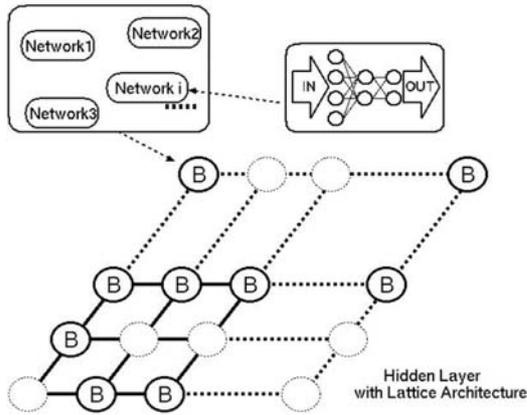
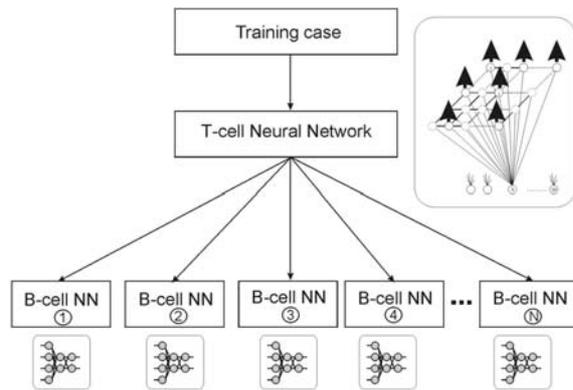Fig. 6.   B-cell neural network.

Fig. 7.   Reasoning process of the trained IMANN.

C. Comparison of the results

Described two methods are based on the different approach like Genetic Programming and Neural Networks. In both methods, however, the entire training data set is divided into two or more subsets, and the learning is performed for each subset. It is interesting that these methods have the common feature in spite of the different approaches.

Table IV shows the comparison of the classification accuracy for the testing dataset between the two methods, which have been trained by using Train_A dataset. IMANN showed better performance than ADG method in classification
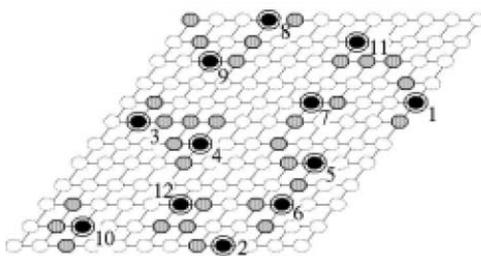

Fig. 8.   Neuron arrangement in hidden layer.

TABLE IV
COMPARISON OF THE CLASSIFICATION ACCURACY.

	rule extraction using ADG	classification using IMANN
Testing dataset	67.8%	82.3%

accuracy. IMANN has an advantage of high classification capability. However, the knowledge on classification is represented by the distributed weights in the network. Therefore, it is necessary to extract comprehensible knowledge from the network.The research for extracting IF_THEN rules from the acquired network also has been performed[7].

On the other hand, the rule extraction method using ADG is inferior to IMANN in classification accuracy, but the acquired rules are comprehensible to users. To improve the classification accuracy, expressive ability should be enriched. It is necessary to revise the GP functions and terminals.

IV. CONCLUSIONS

In this paper, we showed the overview of CHD database and the soft computing methods, which had been previously presented in the data mining competition. Many researches on the data mining from complex and ambiguous databases such as medical databases have already been performed so far. However, each research was performed by using a different database. Therefore, it was difficult to compare the performance of the methods. The development and distribution of the CHD database enables us to make comparison of the performance and to analyze the features of respective methods more objectively. The accumulation of the research results using the CHD database will become profitable data for the data mining domain.

REFERENCES

[1] Machi Suka, Takumi Ichimura, and Katsumi Yoshida, Development of Coronary Heart Disease Database, *Proceedings of 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES'2004)*, LNAI 3214, pp.1081-1088, 2004.
[2] Akira Hara, Takumi Ichimura, Tetsuyuki Takahama, and Yoshinori Isomichi, Extraction of Rules from Coronary Heart Disease Database Using Automatically Defined Groups, *Proceedings of 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES'2004)*, LNAI 3214, pp.1089-1096, 2004.
[3] Shinichi Oeda, Takumi Ichimura, and Katsumi Yoshida, Immune Multi Agent Neural Network and Its Application to the Coronary Heart Disease Database, *Proceedings of 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES'2004)*, LNAI 3214, pp.1097-1105, 2004.
[4] Koza, J., *Genetic Programming, On the Programming of Computers by means of Natural Selection*, MIT Press, 1992.
[5] Dawber, T.R., Kannel, W. B., Revotskie, N., Stokes, J.3rd., Kagan, A., and Gordon, T., Some factors associated with the development of coronary heart disease: six year's follow-up experience in the Framingham Study, *American Journal of Public Health*, Vol.49, pp1349-1356, 1959.
[6] T. Kohonen, Self-organizing maps, Springer Series in Information Sciences, Vol.30, 1995.
[7] Takumi Ichimura, Shinichi Oeda, Machi Suka, Akira Hara, Kenneth J. Mackin, and Katsumi Yoshida, Knowledge Discovery and Data Mining in Medicine, *Advanced Techniques in Knowledge Discovery and Data Mining*, Pal, Nikhil; Jain, Lakhmi C. (Eds.), pp.177-210, 2005.