Evolutionary Hypernetwork Models for Aptamer-Based Cardiovascular Disease Diagnosis

Jung-Woo Ha
Biointelligence Lab.
School of Comp. Sci. & Eng.
Seoul National University
Seoul 151-744, Korea
+82-2-880-1847
jwha@bi.snu.ac.kr

Jae-Hong Eom
Biointelligence Lab.
School of Comp. Sci. & Eng.
Seoul National University
Seoul 151-744, Korea
+82-2-880-1847
jheom@bi.snu.ac.kr

Sung-Chun Kim
GenoProt Inc.
2FL Saeseoul B/D 94-1
Guro 6-dong, Guro-gu,
Seoul 152-841, Korea
+82-2-862-9956
kimgp@genoprot.com

Byoung-Tak Zhang
Biointelligence Lab.
School of Comp. Sci. & Eng.
Seoul National University
Seoul 151-744, Korea
+82-2-880-1833
btzhang@bi.snu.ac.kr

ABSTRACT

We present a biology-inspired probabilistic graphical model, called the hypernetwork model, and its application to medical diagnosis of disease. The hypernetwork models are a way of simulated DNA computing. They have a set of hyperedges representing a subset of features in the training data. These characteristics allow the hypernetwork models to work similarly to associative memories and make their learning results more understandable. This comprehensibility is one of main advantages of the models over other machine learning algorithms such as support vector machines and artificial neural networks which are used in a wide range of applications but are not easy to understand their learning results. Since medical applications require both competitive performance and understandability of results, the hypernetwork models are suitable for this kind of applications. However, ordinary hypernetwork models have limitations that hyperedges cannot be changed after they are sampled once. To improve this diversity problem, we adopted simple evolutionary computation method, the hyperedges replacement strategy as the method of keeping the diversity into conventional hypernetworks in addition to error correction for model learning. To show the improvement, we used aptamer-based cardiovascular disease data. Experiment results show that the hypernetworks can achieve fairly competitive performance and the results are also comprehensible.

Categories and Subject Descriptors

I.5.2 [Computing Methodology, Pattern Recognition, and Design Methodology]: Classifier design and evaluation

General Terms: Algorithms, Experimentation.

Keywords: Hypernetwork, Hypergraph, Aptamer, Cardiovascular disease, Diagnosis, Evolutionary computation.

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

1. INTRODUCTION

Since DNA computing was suggested by Adleman [1], it has been recognized as biology inspired new computational paradigm and motivated many variational methods for different problems [1]. However, there have been many difficulties in implementing DNA computing *in vitro*. For example, the set of experimental constraints including temperatures, molecule densities, and salt concentrations should be controlled strictly for the precise simulation of *in vitro* reactions.

Many alternative *in silico* simulation methods have been suggested to resolve these limitations and to render *in vitro* experiment. The probabilistic library model (PLM) [14] and the hypernetwork models [12] are representative examples of these simulated DNA computing approaches.

The hypernetwork model is a novel random probabilistic graphical models based on undirected graphs. A hypernetwork is a hypergraph which consists of weighted hyperedges. A hyperedge can connect to more than two vertices while the usual edge in conventional graphs connects two vertices. Since hyperedges are made by sampling the features of given observed data randomly, similarly to library elements in PLM, each hyperedge contains the partial contents of given training data. Therefore, whole information of given training data can be reconstructed by retrieving and combining a set of hyperedges. This mechanism is similar to the working method of associative memories. The main advantage of hypernetwork models is that they provide descriptive way of explaining the relation between features and class labels of training data. Another advantage is that they can have competitive performances in diverse pattern recognition problems such as handwritten digit recognition problem [12]. These are advantages of hypernetwork models over other machine leaning algorithms considering that support vector machines (SVM) and artificial neural networks are widely used but are not easy to understand its learning results, and decision trees provide the easy-to-understand results but occasionally show poor classification performances. Therefore hypernetwork models are useful in bioinformatics and medical applications where the comprehensibility of the learning results is as important as the ability to classify and predict the data. However, ordinary hypernetwork models have limitations for diversity. Although the weight of a hyperedge is updated by error correction, hyperedges themselves, which a hypernetwork consists of, cannot be changed after being sampled once. To keep the diversity, we adopt the method of replacing hyperedges as evolu-

tionary computation into an ordinary hypernetwork. We perform cardiovascular disease level prediction using aptamers to show the effect of keeping the diversity of the hypernetwork.

Aptamers are emerged as rival molecules in medical fields providing many advantages over synthetic antibodies recently [4]. Aptamers consist of oligonucleotide sequences which have the capacity to recognize any class of target molecules with high affinity and specificity. For that reason, they are used in a wide range of recent applications including synthetics, biosensors, and material detection, etc. [2] [3] [7] in addition to clinical applications.

In this paper, the classification performances of hypernetworks were compared with several popular machine learning methods including SVMs, decision trees (DTs), and Bayesian networks (BNs). We show that the evolved hypernetwork performs well for the problem of cardiovascular disease level prediction with competitive performances compared with the several popular machine learning methods.

The rest of this paper is organized as follows: In Section 2, we present the basic concept and learning method of the hypernetwork model. The error correction and evolution methods are described in Section 3. In Section 4, we present the basic concept of aptamers and aptamer-biochip data of cardiovascular disease used for the experiments. Experimental results and their analysis are described in Section 5. Concluding remarks and future research are drawn in Section 6.

2. THE HYPERNETWORK MODELS

2.1 The Basics of the Hypernetwork Models

Before referring to the hypernetwork models, we need to define a hypergraph. A hypergraph G is undirected graph with edges which can be connected to more than two vertices i.e. $G = (V, E)$ where V is a set of vertices and E is a set of edges such that $V = \{v_1, v_2, \ldots, v_n\}$, where v_i is a pair of index and binary value, $v_i =$ (index, value), $E = \{E_1, E_2, \ldots, E_m\}$, and $E_i = \{v_{i1}, v_{i2}, \ldots, v_{ik}\}$. E_i is a hyperedge which is different from an edge of a conventional graph for the number of connected vertices. Ordinary edges of a graph connect to at most two vertices, but hyperedges can connect to more than two vertices. A hyperedge of cardinality k is called a k-cardinality hyperedge. Figure 1 shows a hypergraph with eight vertices and four hyperedges.

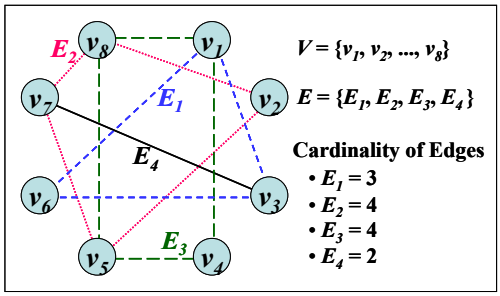

Figure 1. Hypergraph with 8 vertices and 4 hyperedges

A hypernetwork is a specific case of hypergraph. When a weight value is assigned to a hyperedge, we call it a hypernetwork. That is, a hypernetwork is a hypergraph with weighted hyperedges. Formally we define a hypernetwork by using a set of vertices, edges and weights as $H = (V, E, W)$, where $V = \{v_1, v_2, \ldots, v_n\}$, $E = \{E_1, E_2, \ldots, E_m\}$, $E_i = \{v_{i1}, v_{i2}, \ldots, v_{ik}\}$, and $W = \{w_1, w_2, \ldots, w_m\}$. If every hyperedge E_i in E of a hypernetwork H has cardinality k, then we call it a k-uniform hypernetwork.

Hypernetwork models are strongly related to the probabilistic library models (PLM) [6][14] which are simulated DNA computing models. Compared with PLMs, a hypernetwork corresponds to an entire library, a hyperedge to a library element, a vertex to a gene, and a weight to the number of copies of a library element. We can consider PLMs as genotypes of hypernetwork models and hypernetwork models as phenotypes of PLMs. Shortly, the principle of PLMs is adopted in hypernetwork models.

2.2 Hypernetworks as Associative Memories

Associative memories are storage devices which return stored contents from partial contents. Similarly, hypernetwork models play the comparable role of associative memories through the mechanism of storage and retrieval of given data. Compared with associative memories, hyperedges correspond to partial contents and the cardinality of hyperedges is the quantity of the partial contents. When the cardinality of a hyperedge is lower, the hyperedge is probable to be matched to more data. On the other hand, the higher is the cardinality, the more specific is the hyperedge to data that have similar patterns. Therefore, the hypernetwork with lower cardinality behaves as globalist and higher one acts as localist. Figure 2 shows the process of constructing hyperedges from observed data, retrieving them, and classifying unobserved data.

To present the procedure of storage and retrieval as associative memories, we need to define several terms. If C, $C = \{c_1, c_2, \ldots, c_n\}$, is a set of class labels and M, $M = \{m_1, m_2, \ldots, m_n\}$, is a counter set whose ith element is the number of hyperedges with class label c_i, then the contribution of a feature is given by

$$P(c_i \mid X_j) = \frac{m_i}{\sum_{k=1}^{n} m_k}$$

where X_j denotes the vector of features of the jth data.

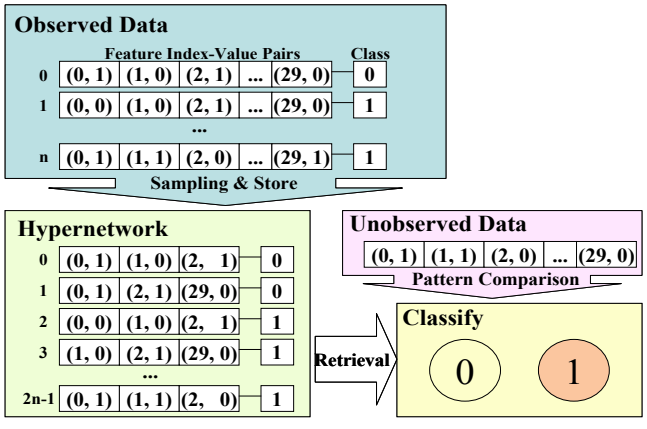

Figure 2. The mechanism of storage and retrieval.

The mechanism of storage and retrieval is as follows:

1. Make hyperedges and store them in a hypernetwork by sampling several index-value pairs of features randomly for an observed datum. The number of pairs is the cardinality of the

hyperedge. The class of a hyperedge is assigned to the class label of the original datum.

2. Repeat Step 1 for an observed datum in several times. The number of repeated sampling is referred to sampling rate.

3. Repeat Step 2 for all observed data set. The number of hyperedges in a hypernetwork is equal to observed data set size by sampling rate.

4. Retrieve a hyperedge E_i, where $E_i = \{v_{i1}, v_{i2}, \ldots, v_{ik}\}$, $v_{ij} = (d_{ij}, x_{ij})$, and compare x_{ij} with the feature value of an unobserved datum whose index is d_{ij}. If an unobserved datum has same index-value pairs of features as all elements of E_i and the class label of E_i is c_k, kth element of C, then increase m_k, the *kth* element of M.

5. Repeat Step 4 for all hyperedges.

6. Calculate $P(c_i|X)$, where X is the vector of features of the unobserved data and c_i is the ith element of C, for all class labels.

7. Predict the class label of the datum with c_i, which satisfying $argmax\ P(c=c_i|X)$.

8. Repeat Step 5, 6, and 7 for all unobserved data.

In above steps, step 1 is the process of storing partial contents and steps from 4 to 7 are the process of returning contents. Hypernetwork models show good classification performance as associative memories for pattern recognition problems including hand-written digit recognition [12].

3. EVOLUTIONARY HYPERNETWORKS

To improve naïve hypernetwork models, we added simple learning and evolutionary computing mechanism. Before the mention of learning and evolving, we need to define some terminology that will be used to explain the way to improve. The ith element T_i of training data set T and the jth element D_j of test data set D with m features which have binary value, and a class label which is element c of class label set C, $C = \{c_1, c_2, \ldots, c_n\}$, is defined as a set which is given by

$$T_i = \{f_{i1}, f_{i2}, \ldots f_{im}, c_i\},$$

$$D_j = \{f_{j1}, f_{j2}, \ldots f_{jm}, c_j\}.$$

According to the procedure constructing a hypernetwork $H = (V, E, W)$ from given data set, a set of vertices V, $V = (v_1, v_2, \ldots, v_k)$, is a subset of a set of features F, $F = \{f_1, f_2, \ldots, f_m\}$, and both v and f are a pair of (index, value). In other words, the arbitrary element v of vertices set V corresponds to a specific element f of feature set F. Therefore the ith k-cardinality hyperedge in a hypernetwork H which has n hyperedges is defined as followed,

$$E_i = \{v_{i1}, v_{i2}, \ldots\ v_{ik}, c_i\},$$

$$W = \{w_1, w_2, \ldots, w_n\}$$

where w_i is the weight of E_i.

The learning and evolving procedures of the hypernetwork are presented in the following subsection.

3.1 Introducing Error Correction

The performance of hypernetwork can be improved by introducing heuristics based error correction procedures. This error correction is to update the weight of hyperedges using the result of learning training data set. The error correction includes following steps:

1. Divide data into training data set and test data set

2. According to procedure explained in 2.2, make a hyper network H and initialize the weights to W_c.

3. For all training data, repeat followed sub-step, where the ith training data T_i.

   1) Select the jth hyperedge E_j of H.

   2) In case of $T_i - \{c_i\} \supset E_j - \{c_j\}$,
   If $c_i = c_j$, then $w_j = w_j \times \delta_p$ ($\delta_p > 1$),
   Otherwise, $w_j = w_j \times \delta_n\ (0 < \delta_n < 1)$.

   3) Repeat 1) and 2) for all hyperedges for T_i.

4. Build a new hypernetwork *H'* with updated weights from Step 3.

5. Through the followed sub-step, estimate classification performance for training data set and test data set.

   1) Select a datum D_i from training (test) data set, where D_i is the ith element of data set.

   2) In case of $D_i - \{c_i\} \supset E_j - \{c_j\}$ and $c_j = c_k$, where E_j is the jth element of hyperedge set E and c_k is the kth element of class label set C, add the weight w_j of to m_k .

   3) Repeat 2) for all hyperedges.

   4) Calculate $P(c_k|X_i)$, where X_i is the vector of features of D_i and c_k is the kth element of C, for all class labels.

   5) Classify the class label of D_i into c_k satisfying

   $argmax\ P(c=c_k|X_i)$.

   6) If $c_i = c_k$, then count it.

6. Save the hypernetwork H which has the highest classification accuracy.

7. Until the termination condition is satisfied, repeat Steps from 3 to 6.

In Step 3, δ_p and δ_n play the role of learning rate. The nearer to 1 are δ_p and δ_n, the slower is learning. This step is error correction and updated weights cause to improve the classification performance of hypernetwork models. Generally, the termination condition is the number of iterations. Compared with DNA computing, updating the weight of hyperedges means that a DNA sequences are selected and amplified by polymerase chain reaction (PCR). By error correction, therefore, learning a hypernetwork is searching optimal combination of weights.

3.2 Evolving Hypernetworks

In addition to learning through error correction, we can further improve hypernetwork models by adopting evolutionary concept. We used simple population replacement strategy for bad solutions

as evolution methods. The goal of evolving hypernetworks is selecting the feature whose makes the classification performance of a hypernetwork higher by assigning its classification accuracy to fitness function. Before we describe the process of evolving hypernetwork, some terminologies need to be defined.

A subgraph S is defined a kind of hypernetworks and is an individual of population, where $S = \{S_1, S_2, \ldots, S_n\}$ and $S_k = (V, E, W)$. That is, S is a small hypernetwork. When a hypernetwork H consists of n subgraphs, H can be represented as

$$H = \bigcup_{k=1}^{n} Sk \ .$$

A subgraph S is individual to be evolved and the size of subgraph set S is population size. A hypernetwork H is made up of subgraphs that have good quality. Unlike 3.1, data set is divided into three parts, training, validation, test data set. The reason that use validation set is that classification accuracy for them is assigned to the fitness function of S. the fitness function of the kth subgraph is called f_k. We replace the subgraphs which has poor fitness function with new subgraphs to keep the diversity. The procedure of evolving hypernetwork models is followed in detail.

1. Divide data set into training, validation, and test data set.
2. Make subgraphs S from training and validation data set.
3. Learn S by using training data for some iterations.
4. Assign classification accuracy for validation data set of S_k to the fitness function f_k
5. Make a modified hypernetwork H' by followed simple evolutionary computation methods.
   1) Selection: execute Step 3 and 4.
   2) Reproduction: Reproduce the specific proportion of subgraphs with good fitness function.
   3) Variation: To keep the diversity, drop the rest which are not selected in 1) and make new subgraphs as size of dropped subgraphs.
   4) Reconstruction: Construct a modified hypernetwork by merging the result of Step 2)
6. Until the stop condition is satisfied, repeat Step 5.
7. Learning the result of Step 6 by error correction which is explained in Section 3.1.

A hypernetwork with hyperedges, which are selected through above evolutionary steps, is probable to classify the data better than one sampled randomly once. That is, evolutionary computation methods such as selection, variation, and reproduction play the role of constructing a hypernetwork with hyperedges including important features.

4. APTAMER-BASED BIOCHIP DATA OF CARDIOVASCULAR DISEASE

4.1 Aptamers

Synthetic antibodies have been used as the most popular class of molecules with capacity of molecular recognition for a wide range of applications such as diagnosis and therapeutics for a few decades. However, aptamers are emerging as rival molecules against antibodies and they have some advantages over antibodies [4]. Aptamers are the oligonucleotide sequences which have the capacity to recognize any class of target molecules with high affinity and specificity. Aptamers are made by the process called the systematic evolution of ligands by exponential enrichment (SELEX) which is made up of selection and amplification steps [4]. Recent researches report that aptamers are used in a wide range of applications including biology and chemical industry beside disease diagnosis and therapeutics. In recent researches, for example, aptamers are applied for RNA interference (RNAi) study with aptamer-siRNA [2][7], detection of biological threat agents [3], and alternative anticoagulants synthesis [10].

4.2 Cardiovascular Disease

Cardiovascular disease (CVD) is a disease affecting the heart or blood vessels. Any abnormal condition characterized by the dysfunction of the heart or blood vessels such as arteriosclerosis, rheumatic heart disease and systemic hypertension. In general, CVD include arteriosclerosis, coronary artery disease, heart valve disease, arrhythmia, heart failure, hypertension, orthostatic hypotension, shock, endocarditis, diseases of the aorta and its branches, disorders of the peripheral vascular system, and congenital heart disease. In affluent western societies such as the USA and Australia, CVD is the severe disease of leading cause of death.

4.3 Data Preparation and Preprocessing

CVD is divided into 3 classes which are stable angina (SA), unstable angina (UA), and myocardial infarction (MI) in the order of its development. It is assumed that CVD patients have the disease-specific and disease level-specific proteins in their blood. To detect these blood-contained proteins, we discovered about 3,000 aptamers by applying SELEX for serum refined from 135 CVD patient bloods including normal group and disease patients. These aptamers having high affinity and specificity to the disease-specific proteins are selected by SELEX. Then 3K aptamer-array biochips are crafted with these selected aptamers. By reacting with 135 patient blood samples we obtained mass protein expression data. Table 1 shows the simple statistics of this data.

Table 1. The statistics of CVD data

Class	Normal	SA	UA	MI	Total
Training	24	22	17	16	79
Validation	8	8	6	6	28
Test	8	8	6	6	28
Total Sum	40	38	29	28	135

The 3K protein expression data have 3,000 feature dimensions and each feature, which is the level of reaction between aptamers and protein, has a real-valued expression level of the range from 0 to 1. This data set was obtained by applying general preprocessing steps including scanning, quality controls, imputation, linear normalization, and log transformation [5]. After these procedures, we selected 150 features, which have high significant values, from total 3000 features with gain-ratio [9] to reduce feature dimension.

Lastly, we discretized each feature value as a binary form for the application of hypernetwork model by applying Fayyad and Irani's binary discretization algorithm in Weka [11]. This is due to the limitation of current hypernetwork model which can only adopt binary-valued feature vectors.

For hypernetwork learning and evolving, we divided data set having 135 patient samples into 3 parts including training, validation, and test set. The validation data set are used to calculate the fitness function when evolving a hypernetwork, and they are merged with training data for learning a hypernetwork by error correction. The each portion of validation and test data is about 20% of entire data set per classes as explained in Table 1.

5. SIMULATION RESULTS

5.1 Experiment Settings

The δ_p and δ_n are the positive and the negative learning rate of hypernetwork learning respectively. We assigned positive learning rate δ_p to 1.0001 to prevent the weight accumulation overflow through iterative learning. However, we maintained flexible negative learning rate to enable wider solution space search. Because it could be supposed that if improved rates by a learning epoch get smaller, the weight of hyperedges get nearer to optimal value for classification. The rule of δ_n is given by

$$
\begin{aligned}
&\text{if } \Delta E \geq 0.01 && \text{then } \delta_n = 0.95\\
&0.005 \leq \Delta E < 0.01 && \text{then } \delta_n = 0.99\\
&0.002 \leq \Delta E < 0.005 && \text{then } \delta_n = 0.995\\
&0.001 \leq \Delta E < 0.002 && \text{then } \delta_n = 0.999\\
&0.0005 \leq \Delta E < 0.001 && \text{then } \delta_n = 0.9995\\
&\Delta E < 0.0005 && \text{then } \delta_n = 0.9999
\end{aligned}
$$

where E_k is the classification error for training data set in the kth epoch and $\Delta E = E_{k-1} - E_k$.

To prevent fast zero convergence of model weights, we assigned the negative learning rate to value slightly less than 1.0. The fast zero weight convergence could leads overfitting of model training. To evolve a hypernetwork, we doubled population size of subgraphs and selected the good half of all subgraphs sorted by their validation accuracy. The stop condition of evolution was set as 10 iterations.

Two data sets are used for model simulation. One is Wisconsin diagnostic breast cancer data set, and the other is aptamer-based CVD data set. Since hypernetwork models have been applied for the problems with binary feature value such as digit recognition and text mining, we used WDBC data in addition to CVD data to show that the hypernetwork models can also be good at classifying the problem with real-value features through proper binary discretization. The experimental results with these two data sets are presented in the following sections.

5.2 Wisconsin Diagnostic Breast Cancer Data

Data set having real value features, Wisconsin Diagnostic Breast Cancer (WDBC) data, was collected from UCI machine learning repository for hypernetwork-based classification. The WDBC data have binary class label which is either malignant or benign, 569 samples that consist of 212 malignant and 357 benign samples with 30 real-valued features. The feature value of the WDBC data was also converted to binary data by preprocessing.

To compare hypernetwork models with other machine learning algorithms, we used the implementation of decision trees (J48), SVM (SMO), Bayesian networks in Weka and simulated an ordinary hypernetwork (O-HN) and an evolved hypernetwork (E-HN). The polynomial kernel is used in SVM and K2 is used as search algorithms in Bayesian networks. An ordinary hypernetwork means model that classifies by basic operation of storage and retrieval. We ran each algorithm nine times with test data having 20% of original data and averaged these results. The cardinality of the hypernetworks was set to 3, and sampling rate was set to 50. Furthermore, in case of evolved hypernetworks, population size was assigned to 4. Table 2 shows the classification accuracy of each algorithm for test data set. The p–value was calculated through t–test. The accuracy of hypernetwork of the table is maximum value throughout all testing epochs.

Table 2. The classification results for WBCD data set

Epochs	Classification accuracies				
	DT	**SVM**	**BN**	**O-HN**	**E-HN**
1	93.860	93.86	92.110	83.478	**98.261**
2	95.614	98.246	98.246	86.957	**93.043**
3	93.86	96.491	93.860	88.696	**95.652**
4	89.474	90.351	91.228	83.478	**94.783**
5	93.860	93.86	92.983	84.378	**93.913**
6	85.965	89.474	87.719	84.378	**96.522**
7	92.105	93.86	92.983	81.739	**97.391**
8	92.983	97.368	95.614	83.478	**96.522**
9	94.737	96.491	95.614	83.478	**97.391**
Average	**92.495**	**94.445**	**93.373**	**84.451**	**95.942**
p-value	**0.005**	0.111	**0.023**	**0.000**	–

In above results, all algorithms except ordinary hypernetworks present good classification and the accuracy of the evolved hypernetwork model is the best of them. Especially, considering p-value by t-test, we can insist that the hypernetwork model should not only be improved significantly but evolved hypernetworks can also classify WDBC problem better than decision tree and Bayesian networks within 5% confidence interval. Consequently, hypernetwork models can classify the problem with real-valued features though the values of vertices are restricted to binary values.

5.3 Aptamer-based CVD Data

Same as Section 5.2, we used 5 algorithms to classify the data. Table 3 shows the classification results of these different models.

The result is very similar to the result of Table 2 in the Section 5.2. From the results, it can be thought that the evolved hypernetwork models are superior to not only the ordinary hypernetwork models but also decision trees and Bayesian networks in this classification task.

Table 3. The classification results for CVD data set.

Epochs	Classification accuracies				
	DT	SVM	BN	O-HN	E-HN
1	74.074	77.778	70.37	75	**78.571**
2	66.667	85.185	59.259	67.857	**78.571**
3	55.556	70.37	70.37	75	**82.143**
4	74.074	88.889	62.963	64.286	**82.143**
5	62.693	85.185	74.074	71.429	**78.571**
6	74.074	88.889	74.074	71.429	**85.714**
7	81.482	81.482	74.074	64.286	**78.571**
8	70.37	85.185	81.482	67.857	**82.143**
9	74.074	77.778	66.667	67.857	**82.143**
Average	**70.340**	**82.305**	**70.370**	**69.445**	**80.952**
p–value	**0.0024**	0.17566	**0.001**	**5.8E-06**	–

Figure 3 shows the changes of average training and testing classification accuracies of hypernetworks with cardinality of 4 as model learning proceeds. The population size of model was set to 4 and sampling rate per subgraph was set to 25. The accuracies of Figure 3 are calculated by averaging 10 separate runs. The learning pattern of hypernetwork model in Figure 3 follows the pattern of general machine learning methods.

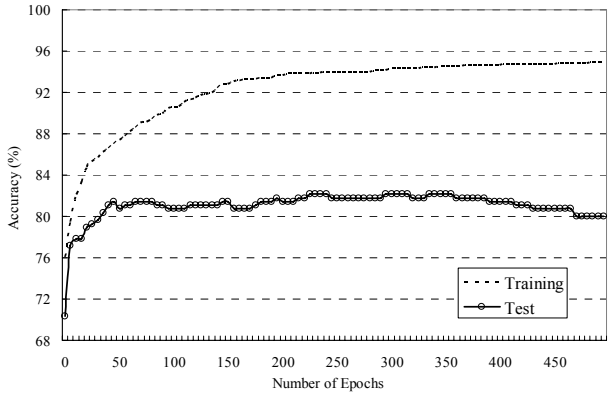

Figure 3. The learning curve for training and testing of hypernetwork models with 4–cardinality

The model training accuracy was increased through all epochs up to about 96% and the model testing accuracy starts to decrease from about 350 epoch after reaching its peak accuracy, which is about 82%, at the range of 300~350 epochs. The fluctuations of accuracy in the range of from 50 to 350 epochs were caused by averaging model accuracies.

Figure 4 shows the effects of changing cardinality of hypernetwork models. To remove the effect of bad accuracy due to lack of hyperedges, we set population size to 4 and sampling rate to 100.

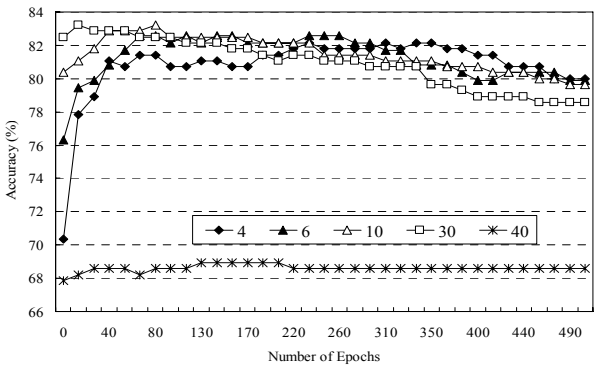

Figure 4. The average accuracies for different cardinalities

Figure 4 shows the average test accuracies of hypernetworks for different cardinalities as model training epoch proceeds. Note that there are difference between the patterns of lower cardinality hypernetworks below 10 and higher ones over 20. In case of hypernetworks with low cardinality, the effect of learning is much bigger than higher ones. Although there are little effects of learning in hypernetworks with higher cardinalities over 20, they can classify the data well by using mechanism of storage and retrieval merely. Models with relatively too high cardinality over 40 show poor classification performances because there are few hyperedges matched to test data.

Table 4 shows the results of disease prediction using hypernetwork model with cardinality of 4. The results are accumulated values for 10 simulations. Accumulated values in each simulation are the number of class label classified for test data by a hypernetwork with best classification accuracy.

Table 4. The disease class prediction results with the hypernetwork model

Class / Predicted	Normal	SA	UA	MI
Normal	**78 (97.25%)**	7 (8.75%)	2 (3.33%)	14 (23.33%)
SA	1 (1.25%)	**73 (91.25%)**	0 (0.0%)	14 (23.33%)
UA	0 (0.0%)	0 (0.0%)	**58 (96.67%)**	5 (8.33%)
MI	1 (1.25%)	0 (0.0%)	0 (0.0%)	**27 (45.0%)**

In Table 4, all classes except class MI were classified with good accuracy. Nevertheless, the ratio of misclassification for MI class is over 50%. Compared with class UA with similar data size to MI, the accuracy for class MI is abnormal significantly. We can guess that the interactions between features are the essential factor in class MI.

Table 5 shows the analysis results of the relations between aptamers and class labels by simulating lower cardinality hypernetworks. The result was collected from the three 3–uniform hypernetworks which have test accuracy over 85%. The pairs of index and value in the table are 10 vertices that appeared the most frequently in all hyperedges of three hypernetworks, and proportion values are the ratio of hyperedges that have the vertices which are

represented with index-value pairs per class labels. The above result can be obtained easily through analyzing hyperedges in case of hypernetworks with lower cardinality below 5. It is remarkable that there are strong relations between index-value pairs of features and class labels. We can suppose that above aptamers have primary effects with CVD diagnosis and react to CVD-related proteins. It is needed that aptamers that seems to be related to CVD is identified to discover CVD-related proteins.

Table 5. The relations between features and class labels

Index	Value	Class	Proportion (%)	Index	Value	Class	Proportion (%)
124	0	**NOR**	**100**	44	0	**NOR**	**96.4**
		UA	0			UA	0
		SA	0			SA	3.6
		MI	0			MI	0
110	1	**NOR**	**85**	63	0	**NOR**	**96.7**
		UA	0			UA	0
		SA	0			SA	3.3
		MI	15			MI	0
127	1	NOR	0	72	1	NOR	0
		UA	**86.3**			UA	24.4
		SA	13.7			**SA**	**68.9**
		MI	0			MI	6.7
91	0	**NOR**	**100**	24	0	**NOR**	**91.5**
		UA	0			UA	0
		SA	0			SA	8.5
		MI	0			MI	0
17	1	NOR	0	101	0	NOR	7.7
		UA	**98.6**			UA	0
		SA	1.4			SA	0
		MI	0			**MI**	**92.3**

6. CONCLUSION

As represented in the above results, hypernetwork models can be improved significantly by hyperedge replacement and error correction to keep the diversity. Compared with other machine learning algorithms, furthermore, evolved hypernetwork models show good classification performance for data with real-valued features. Beside the aspects of classification performance, hypernetwork models allow to understand biological meaning of the simulation results like the relation between features effectively. This is the main advantage of hypernetwork models over machine learning algorithms such as SVM and artificial neural networks which classify data well but are relatively difficult to understand meanings of the result. Therefore, we expect that hypernetwork models can play the role of the useful classifiers and analysis tools in bioinformatics, especially in medical diagnosis applications.

Considering inner parts of hypernetwork models, each of lower and higher cardinality hypernetwork can be applied for different purposes. Generally a higher cardinality about 10%~30% of feature size is better than a 2~5 cardinality hypernetwork with respects to accuracy and overhead. Since higher one can classify well without learning and evolving, there are little overheads for improvement methods compare with lower one. Nevertheless, higher one is too complex to analyze and understand the relation between features that hyperedges consist of. On the other hand, although a lower cardinality hypernetwork has overheads for learning and evolving, it has the advantage of comprehensibility for the result. Regarding the relation between hypernetwork models as simulated DNA computing, a lower cardinality hypernetwork can be implemented easier than higher one in *in vitro* DNA computing.

In the aspects of data analysis, primary aptamers, which have the strongest effect on deciding the class label of data, can be discovered efficiently by using the hypernetwork model. It is needed that interaction between aptamers is found by analyzing the lower cardinality hypernetwork. In addition, CVD related aptamers and corresponding proteins should be identified for the practical use in medical diagnosis.

7. ACKNOWLEDGMENTS

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the NRL Program funded by the Ministry of Science and Technology (No. M10400000349-06J0000-34910), the Ministry of Industry and Commerce through the Molecular Evolutionary Computing (MEC) Project, and the Ministry of Education and Human Resources Development under the BK21-IT Program. The ICT at Seoul National University provided research facilities. Jae-Hong Eom was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2006-511-D00355).

8. REFERENCES

[1] Adleman, L.M., "Molecular computation of solutions to combinatorial problems," *Science*, 266(5187), pp. 1021–1024, 1994.

[2] Davidson, B.L., "All in the RNA family," *Nature Biotechnology*, 24(8), pp. 951–952, 2006.

[3] Iqbal, S.S., Mayo, M.W., Bruno, J.G., Bronk, B.V., Batt, C.A., and Chambers, J.P., "A review of molecular recognition technologies for detection of biological threat agents", *Biosensor & Bioelectronics*, 15(11–12), pp. 549–578, 2000.

[4] Jayasena, S.D., "Aptamers: an emerging class of molecules that rival antibodies in diagnostics," *Clinical Chemistry*, 45(9), pp. 1628–1650, 1999.

[5] Kim, B.-H., Kim, S.-C., and Zhang, B.-T., "Biomarker detection on aptamer-based biochip data by potential SVM," In *Proceedings of the 33th Korea Information Science Society Fall Conference*, 33(2A), pp. 22–27, 2006 (in Korean).

[6] Kim, S., Heo, M.-O., and Zhang, B.-T., "Text classifiers evolved on a simulated DNA computer," *IEEE Congress on Evolutionary Computation* (*CEC 2006*), pp. 9196–9202, 2006.

[7] McNamara, J.O. 2nd, Andrechek, E.R., Wang, Y., Viles, K.D., Rempel, R.E., Gilboa, E., Sullenger, B.A., and Giangrande, P.H., "Cell type–specific delivery of siRNAs with aptamer-siRNA chimeras," *Nature Biotechnology*, 24(8), pp.1005–1015, 2006.

[8] Newman, D.J., Hettich, S., Blake, C.L., and Merz, C.J., UCI Repository of machine learning databases, http://www.ics.uci.edu/~mlearn/MLRepository.html, Irvine, CA: University of California, Department of Information and Computer Science.

[9] Quinlan, J.R., "Induction of decision tree," *Machine Learning*, 1(1), pp. 81–106, 1986.

[10] Rusconi, C.P., Roberts, J.D., Pitoc, G.A., Nimjee, S.M., White, R.R., Quick, G. Jr, Scardino, E., Fay, W.P., and Sullenger, B.A., "Antidote-mediated control of an anticoagulant aptamer in vivo," *Nature Biotechnology*, 22(11), pp.1423–1428, 2004.

[11] University of Waikato New Zealand, "Waikato Environment for Knowledge Ananlysis (Weka)," http://www.cs.waikato.ac.nz/ml/weka/index.html

[12] Zhang, B.-T. and Kim, J.-K., "DNA hypernetworks for information storage and retrieval", *Lecture Notes in Computer Science* (*DNA 12*), 4287, pp. 298–307, 2006.

[13] Zhang, B.-T., "Random hypergraph models of learning and memory in biomolecular networks: shorter-term adaptability vs. longer-term persistency," *The 1st IEEE Symposium on Foundations of Computational Intelligence* (*FOCI '07*), 2007 (to appear).

[14] Zhang, B. –T. and Jang, H.-Y., "A Bayesian algorithm for in vitro molecular evolution of pattern classifiers," *Preliminary Proceedings of the Tenth International Meeting on DNA Computing (DNA10)*, pp.294-303, 2004.