12

APPLYING MACHINE LEARNING TECHNIQUES TO ANALYSIS OF GENE EXPRESSION DATA: CANCER DIAGNOSIS

Kyu-Baek Hwang, Dong-Yeon Cho, Sang-Wook Park, Sung-Dong Kim, and Byoung-Tak Zhang {kbhwang, dycho, swpark, sdkim, btzhang}@bi.snu.ac.kr
Biointelligence Laboratory, School of Computer Science and Engineering, Seoul National University, Seoul 151-742, Korea

Abstract. Classification of patient samples is a crucial aspect of cancer diagnosis. DNA hybridization arrays simultaneously measure the expression levels of thousands of genes and it has been suggested that gene expression may provide the additional information needed to improve cancer classification and diagnosis. This paper presents methods for analyzing gene expression data to classify cancer types. Machine learning techniques, such as Bayesian networks, neural trees, and radial basis function (RBF) networks, are used for the analysis of the CAMDA Data Set 2. These techniques have their own properties including the ability of finding important genes for cancer classification, revealing relationships among genes, and classifying cancer. This paper reports on comparative evaluation of the experimental results of these methods.

Key words: Bayesian networks, neural trees, radial basis function (RBF) networks, gene expression data analysis

INTRODUCTION

The treatment of cancer depends on its type. Accordingly, classification of patient samples is a crucial aspect of cancer diagnosis and treatment. It has been suggested that gene expression may provide the additional information needed to improve cancer classification and diagnosis. Recent technological advances in monitoring gene expression have led to a dramatic increase in gene expression data. Expression chips manufactured using technologies derived from computer-chip production can now measure the expression of

thousands of genes simultaneously. A method for performing cancer classification using gene expression data is presented in [Golub et al., 1999] and [Slonim et al., 2000].

In this paper, we present methods for performing classification of patient samples by gene expression data analysis using machine learning techniques. Data-driven prediction must be able to extract essential features from individual examples and to discard unwanted information. Machine learning techniques are known to be excellent at discarding and compacting redundant information. We apply Bayesian networks, neural trees, and radial basis function (RBF) networks to cancer classification based on gene expression data. Among these methods, Bayesian networks are a kind of probabilistic graphical model. Neural trees and RBF networks are extensions of conventional neural networks and useful for regression.

The CAMDA Data Set 2 involves classifying acute leukemias. Acute leukemias can broadly be divided into two classes, acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). The dataset consists of 38 training samples and 34 test samples. Each sample comes from a patient and has 7,129 attributes that correspond to human gene expression levels. From the machine learning point of view, the cancer classification problem is formulated as follows. First, a model is constructed from training samples using a machine learning algorithm. Then, performance of the model is measured by test sample classification.

Our experimental results show that all training samples are classified correctly. The classification accuracy for test samples is also very high though it varies depending on the applied machine learning techniques. In addition to the classification accuracy, compacting redundant gene expression information and the discovery of relationships among genes are important and useful. In this sense, Bayesian networks are very interesting since they have the ability to disclose the probabilistic relationships among genes. Neural trees are also interesting since they can discover important genes for classification automatically, during the evolutionary process of model building.

This paper is organized as follows. The second section describes cancer classification with Bayesian networks and its experimental results. In the third and fourth sections, descriptions of cancer classification with neural trees and radial basis function (RBF) networks and their experimental results are given. Then, we compare and evaluate the experimental results of three machine learning techniques on the cancer classification problem. Conclusion and future work are presented in the last section.

MINING WITH BAYESIAN NETWORKS

Bayesian Networks

The Bayesian network represents the joint probability distribution for a set of random variables efficiently based on the concept of conditional independence. A Bayesian network assumes a form of directed acyclic graph (DAG). Figure 1 shows an example Bayesian network structure. Each node in the graph corresponds to a random variable and each edge represents the probabilistic dependency between variables. A Bayesian network which consists of n nodes (variables), $\mathbf{X} = \{X_1, \ldots, X_n\}$, represents the joint probability distribution as follows:

$$P(\mathbf{X}) = \prod_{i=1}^{n} P(X_i \mid \mathbf{Pa}_i), \qquad [1]$$

where $\mathbf{Pa}_i$ is the set of parents of X_i in the network structure. $P(X_i \mid \mathbf{Pa}_i)$ is the local probability distribution related to the node X_i. The global structure of a Bayesian network encodes the conditional independence relationships among all variables and is called to be the qualitative part. Local probability distributions for all nodes constitute the quantitative part of the Bayesian network.

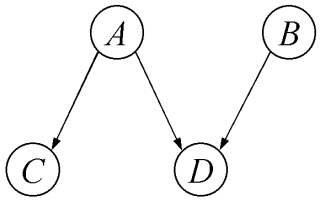

Figure 1. The directed acyclic graph (DAG) structure of an example Bayesian network that consists of four variables (nodes) A, B, C, and D. The joint probability distribution for these variables is represented as $P(A, B, C, D) = P(A)P(B|A)P(C|A, B)P(D|A, B, C)$ by the general chain rule. The network structure encodes the conditional independencies, such as $(A \perp B)$, $(C \perp B|A)$, and $(C \perp D|A, B)$ ($(X \perp Y|Z)$ denotes that X and Y are independent given Z). Using these conditional independencies, the joint probability distribution can be represented more compactly as $P(A, B, C, D) = P(A)P(B)P(C|A)P(D|A, B)$.

Because a Bayesian network encodes the joint probability distribution for a set of variables, the conditional probability of any interesting variable given observations of some of the other variables can be inferred efficiently. Therefore, once the Bayesian network whose nodes represent gene

expression levels and the cancer class label is constructed from the gene expression data, the probability of the cancer class label given some gene expression levels for a new sample can be inferred. This is the Bayesian network classifier. [Jensen, 1996] and [Pearl, 1988] present efficient inference algorithms exploiting the structure of Bayesian networks. In addition to the classification, the edges in the Bayesian network structure denote the possibilities of causal relationships between variables.

Learning Bayesian networks from data is generally NP-hard [Chickering, 1996]. So, heuristic search algorithms, such as greedy search, greedy search with restart, and simulated annealing are used [Friedman and Goldszmidt, 1999].

Applying Bayesian Networks to the Cancer Classification Problem

In the analysis of CAMDA Data Set 2, each gene is regarded as a random variable. Thus, each gene corresponds to a node in a Bayesian network. There is an additional node for the leukemia class label (ALL or AML). The procedure of constructing a Bayesian network classifier for acute leukemias is as follows.

A node X_i in the Bayesian network has its local probability distribution $P(X_i \mid \mathbf{Pa}_i)$ and the local probability distribution model should be determined. In general, the unrestricted multinomial distribution model and the linear regression model are used for the local probability distribution of Bayesian networks. In this paper, the unrestricted multinomial distribution model is used. This model is very expressive. Also, learning algorithms and inference algorithms for the Bayesian network that has the unrestricted multinomial distribution, are established well. Dirichlet distribution is used for the parameters of the unrestricted multinomial distribution [Heckerman, 1996]. Gene expression levels are numerical values and must be discretized for the unrestricted multinomial distribution. In our experiments, all gene expression levels are transformed into two values, that is, 0 (under-expressed) and 1 (over-expressed). Only two gene expression levels are used because of the small size of the training dataset. There are only 38 training samples and the inevitable data sparseness problem can be smoothed by the small number of discretization levels. There are several discretization methods [Dougherty et al., 1995] and the following method was used. Each gene expression level was divided into under-expressed (0) and over-expressed (1) on the basis of its mean expression level across the training samples.

There are 7,129 genes and this number is too large to be applied directly to a general Bayesian network learning algorithm. Moreover, the small

number of training samples available makes the quantitative part (local probability distributions) of the learned Bayesian networks unreliable. In order to learn a reliable Bayesian network from such a small dataset, an appropriate number of genes that are significantly correlated to the cancer classification should be selected. To accomplish this from only 38 training samples, we selected four genes that are significantly correlated to the leukemia class. First, the mutual information value between the leukemia class label and each gene was used to select genes. The mutual information $I(X;Y)$ measures the amount of information that a random variable X contains about another random variable Y [Cover and Thomas, 1991] and is calculated as

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)}. \qquad [2]$$

Here, $p(\cdot)$ denotes the empirical probability estimated from the training dataset and Σ represents the summation over the entire configurations of X and Y. But, the selected four genes with high mutual information values were all highly expressed only in AML and the Bayesian network which consists of these genes showed poor classification performance.

So, two genes were selected from the genes that are highly expressed in ALL and another two genes were selected from the genes that are highly expressed in AML. For this, *P*-metric in [Slonim et al., 2000] was used as well. *P*-metric is defined as follows:

$$P(g,c) = \frac{\mu_1 - \mu_2}{\sigma_1 + \sigma_2}, \qquad [3]$$

where c is the class vector and g is the expression vector. μ_1 and μ_2 are within-class mean expression levels in class 1 and class 2, respectively. σ_1 and σ_2 are standard deviations of expression levels within class 1 and 2, respectively. Table 1 shows four selected genes and their mutual information values and *P*-metric values.

With the selected four genes in Table 1, a Bayesian network was constructed from the training data. The Bayesian network learning procedure consists of two parts. The first part is for structural learning. This part consists of searching for the network structure that best fits the training data. To formalize the goodness of fit of a network structure with respect to the training data, some scoring metric can be used.

Table 1. Four selected genes based on the mutual information value and the *P*-metric value. The GenBank accession number is from http://www.genome.wi.mit.edu/MPR/genes-on-hu6800.html.

	Gene description	*GenBank accession number*	*MI value*	*P-metric value*
Highly expressed in ALL	C-myb gene extracted from Human (c-myb) gene, complete primary cds, and five complete alternatively spliced cds	U22376_cds2	0.37707	1.33931
	MB-1 gene	U05259_rna1	0.31631	1.10318
Highly expressed in AML	Zyxin	X95735	0.70425	-1.40577
	Leukotriene C4 synthase (LTC4S) gene	U50136_rna1	0.50185	-1.421708

There are several such scoring metrics and they are known to be asymptotically the same. Among them, the BD (Bayesian Dirichlet) metric [Heckerman et al., 1995] was used in the experiments. This metric is defined as follows:

$$p(D,B) = p(B)\cdot\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})}\prod_{k=1}^{r_i}\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})}, \qquad [4]$$

where D is the training data and B is a network structure, n is the number of nodes, q_i is the number of possible configurations of parents of node i, and r_i is the number of possible values of node i. α_{ijk} is the Dirichlet prior and encodes the prior knowledge about the domain. Uninformative prior value in [Heckerman et al., 1995], $\alpha_{ijk} = 1$, was used in the experiments. N_{ijk} is the number of cases in the training data D that node i has its k^{th} value and its parents have their j^{th} configuration. α_{ij} and N_{ij} are calculated as

$$\alpha_{ij} \equiv \sum_{k=1}^{r_i}\alpha_{ijk} \qquad N_{ij} \equiv \sum_{k=1}^{r_i}N_{ijk}. \qquad [5]$$

$p(B)$ is the prior score for the network structure B and the same value was assigned for all the structures. $\Gamma(\cdot)$ is the gamma function which satisfies $\Gamma(x + 1) = x\Gamma(x)$ and $\Gamma(1) = 1$. In general, the search space for the structural learning is extremely large. For a Bayesian network with n variables, the size of the search space is about $n! \times 2^{n(n-1)/2}$. However, the Bayesian network in our experiments has only five nodes (four nodes for gene expression levels and one node for the leukemia class) and it is possible to find out the best structure in the entire search space. We also restricted the number of parents of each node to be less than three in the structural search procedure to get reliable local probability distributions with such a small dataset.

The second part of Bayesian network learning is parametric learning for the local probability distributions. With Dirichlet distribution for the parameters of the unrestricted multinomial distribution, the parametric learning is resolved by a simple calculation [Heckerman, 1996]. Figure 2 is the best Bayesian network structure containing the four genes shown in Table 1. It takes a few minutes to construct this Bayesian network on a Pentium II machine.

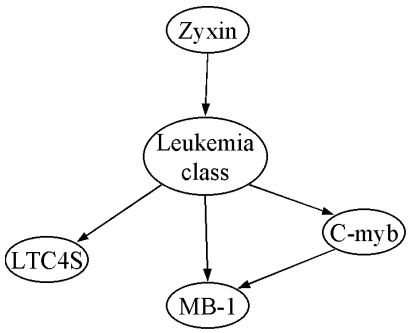

Figure 2. The Bayesian network structure with four genes shown in Table 1. An edge in the Bayesian network represents the probabilistic dependency between the child node and its parent node. This indicates the possibility of causal relationships between the child and parent. The local probability distribution for each node is not shown here.

To classify the cancer, the conditional probability *P*(Leukemia class | C-myb, MB-1, Zyxin, LTC4S) is inferred from the Bayesian network shown in Figure 2. The probability of 0.5 is regarded as misclassification.

Experimental Results for Bayesian Networks

The Bayesian network in Figure 2 correctly classifies all training samples and misclassifies only two test samples. The misclassified two samples in the test dataset are the 66^{th} and 67^{th} samples. Also, the network structure represents the probabilistic relationships among the four genes and the leukemia class. In Figure 2, an edge denotes the direct probabilistic dependency between the child node and its parent node. For example, “Leukemia class” depends on “Zyxin” probabilistically and vice versa. In addition, this probabilistic dependency indicates the possibility of causal dependency. It is useful to analyze gene expression patterns by Bayesian network learning to facilitate causal analysis.

MINING WITH NEURAL TREES

Neural Trees

Neural tree models [Zhang, 1994] represent multilayer feedforward neural networks as tree structures. They have heterogeneous neuron types in a single network, and the connectivity of the neurons is irregular and sparse [Zhang et al., 1997]. There are two types of neurons used in typical neural trees. One is the Σ neuron that computes the weighted sum of inputs. The other is the Π neuron that computes the product of weighted inputs. A neural tree approximates the functional relationship between inputs and outputs by its structure and weights. The advantage of neural trees over conventional neural networks is their flexibility. Neural trees can represent more complex relationships than neural networks and permit structural learning and automatic feature selection. Figure 3 is an example neural tree. In this figure, x_i is the value of the i^{th} input node and w_{ij} is the connection weight to node i from its j^{th} input.

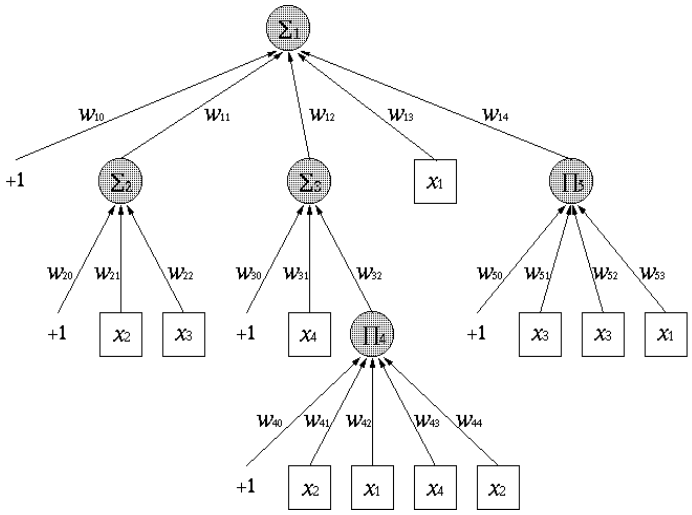

Figure 3. An example neural tree. This neural tree has four input variables (x_1, x_2, x_3, x_4). The terminal nodes denote input nodes. Note that some input nodes appear more than twice. w_{ij} is the synaptic weight to node i from its j^{th} input and +1 is the bias to the neuron. Non-terminal nodes represent Σ neurons and Π neurons and the top-level circular node (node 1) denotes the output of the neural tree.

Evolutionary algorithms such as genetic programming can be used to find the appropriate structure and weights of neural trees. The Bayesian evolutionary algorithm (BEA) [Zhang and Cho, 2001] was used in the

experiments. BEA allows the background knowledge about the given data, which is represented by the prior probability, to be incorporated in the evolutionary procedure. Thus a local search can be performed more effectively without overfitting neural trees to the training data.

Applying Neural Trees to the Cancer Classification Problem

All gene expression levels are fed to the neural tree as input. The output of the neural tree corresponds to the leukemia class label (ALL or AML). Through a structural learning process, essential genes for cancer classification are included into the neural tree and less important genes are weeded out automatically. Moreover, contrary to the Bayesian network, there is no need to discretize the inputs for neural trees. In many cases, discretization reduces information contained in the original data. In neural tree learning, all gene expression levels were linearly scaled into the interval [0.01, 0.99]. For the output value of neural tree learning, ALL was set to 0.01 and AML was set to 0.99.

The Bayesian evolutionary algorithm (BEA) [Zhang and Cho, 2001] was used to construct neural trees from training samples. It took about 10 hours to learn a neural tree from training data on a Pentium II machine.

Experimental Results for Neural Trees

Ten neural trees were constructed by BEA and their performance is summarized in Table 2. Because the evolutionary process for neural tree learning is a probabilistic search, the learned neural trees can be different from each other. Experimental results show that training samples can always be separated into two classes perfectly. Test samples are also separated well with only one misclassification for the best case although the average number of misclassified test samples is about 5. The only misclassified sample in the best case was the 66^{th} sample.

The best model found in the second run uses just 16 genes among 7,129 genes to classify cancer types. The other genes were automatically weeded out through the evolutionary process used in neural tree learning. Table 3 shows these 16 genes. Among the 16 genes, only “Zyxin” and “GB DEF = Homeodomain protein HoxA9 mRNA” were found by other gene selection methods ([Golub et al., 1999] and [Slonim et al., 2000]).

Table 2. Prediction accuracy of neural trees. Training error is the regression error and nearly zero in all cases. The neural tree found in the second run shows the best classification performance on the test dataset.

Run	*Training error*	*Test error*
1	5.09E-04	5/34
2	4.53E-04	**1/34**
3	4.84E-04	9/34
4	8.13E-04	5/34
5	4.37E-04	8/34
6	6.35E-04	2/34
7	5.40E-04	4/34
8	4.83E-04	8/34
9	5.69E-04	2/34
10	7.43E-04	5/34
Average	5.67E-04	4.9/34

Table 3. The 16 genes found by the best neural tree. Only "Zyxin" and "GB DEF = Homeodomain protein HoxA9 mRNA" are also found by other gene selection methods. With these 16 gene expression levels, the neural tree classifies all the training samples correctly and misclassifies only one test sample. The GenBank accession number in parentheses is from http://www.genome.wi.mit.edu/MPR/genes-on-hu6800.html.

Type I keratin, hHa5 (X90763)	**Zyxin (X95735)**
GB DEF = Homeodomain protein HoxA9 mRNA (U82759)	MST1R Protein-tyrosine kinase RON (X70040)
Fibroblast Growth Factor Receptor K-Sam, Alt. Splice 1 (HG3432-HT3618)	Clone 23803 mRNA, partial cds (U79298)
PROBABLE UBIQUITIN CARBOXYL-TERMINAL HYDROLASE (D29956)	MYL1 Myosin light chain (alkali) (M31211)
LEUKOCYTE ELASTASE INHIBITOR (M93056)	Nuclear Factor Nf-Il6 (HG3494-HT3688)
GB DEF = Cdc5, partial cds (D85423)	KIAA0159 gene (D63880)
LTB4H Leukotriene B4 omega hydroxylase (cytochrome P450, subfamily IVF) (D12620)	ATP6A1 ATPase, H+ transporting, lysosomal (vacuolar proton pump), alpha polypeptide, 70kD, isoform 1 (L09235)
RING protein (Y07828)	VIL2 Villin 2 (ezrin) (X51521)

MINING WITH RADIAL BASIS FUNCTION (RBF) NETWORKS

RBF Networks

Radial basis function (RBF) networks have a similar structure to that of neural networks, but their hidden neurons contain RBFs, a statistical transformation based on a Gaussian distribution. Figure 4 represents an RBF network structure. In Figure 4, x_i is the i^{th} input node and z_i is the i^{th} hidden node whose output value is computed as

$$z_i = \exp\left(-\frac{\|\mathbf{c}_i - \mathbf{x}\|^2}{w_i^2} \right) \quad [6]$$

Here, $\mathbf{c}_i$ is the m-dimensional center of z_i, w_i is the width of z_i, and $\mathbf{x}$ is the m-dimensional input vector fed to z_i. y is the output of an RBF network and computed by the following equation:

$$y = \sum h_i z_i + h_0. \quad [7]$$

RBF networks are useful for local approximations to nonlinear input-output mapping [Haykin, 1999]. The resource-allocating network (RAN) algorithm [Platt, 1991] has been used for learning the RBF network and the appropriate number of hidden neurons is automatically determined through the learning process in this algorithm. In our experiments, the active RAN algorithm, the modified version of the RAN algorithm for the active learning scheme was used [Park and Zhang, 2000].

Applying RBF Networks to the Cancer Classification Problem

An appropriate number of genes are selected based on the *P*-metric [Slonim et al., 2000] and mutual information values. And then, these gene expression levels are fed into the RBF network as input. Similar to the case of neural tree learning, all input gene expression levels were linearly scaled into the interval [0, 1]. The output value of an RBF network corresponds to the leukemia class label and was set to 0 for ALL and 1 for AML.

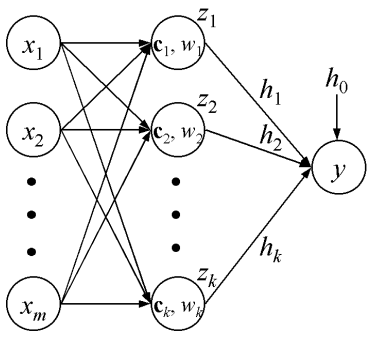

Figure 4. A radial basis function (RBF) network. (x_1, x_2, …, x_m) is the m-dimensional input vector. z_1, z_2, …, z_k are k hidden neurons. The number of hidden neurons are determined automatically through the learning process. $\mathbf{c}_i$ and w_i are the center and the width of the hidden neuron z_i, respectively. h_i is the weight of z_i and h_0 is the bias to the output y.

Experimental Results for RBF Networks

Experiments were performed on 3 different sets of input genes. First, all 7,129 genes were used. Second, 50 gene expression levels by *P*-metric [Slonim et al., 2000] were used. On the third experiment, 10 genes were selected by mutual information and used for learning.

The active RAN (ARAN) algorithm [Park and Zhang, 2000] was used to construct RBF networks from training samples. In the case of using all 7,129 gene expression levels as an input, it took about 12 hours to learn an RBF network on a Pentium II machine. In the second and third experiments where 50 and 10 genes were used for training, it took a few minutes to construct an RBF network. Because the active RAN algorithm has the probabilistic property, its result may be different according to each run. So, we ran the active RAN algorithm on the second and the third set of genes ten times respectively.

Table 4 shows the result of each experiment. The RBF network with 50 gene expression levels classifies all training samples correctly and its test error is 1.3/34 in average.

Table 4. The prediction accuracy of RBF networks. The RBF network constructed with 50 genes selected by the *P*-metric value shows the best performance.

	All genes	*50 genes*	*10 genes*
No. of runs	1	10	10
Average training error	0/38	0/38	0/38
Average test error	4/34	**1.3/34**	2.5/34

COMPARISON AND EVALUATION OF THE RESULTS OF MACHINE LEARNING TECHNIQUES ON THE CANCER CLASSIFICATION PROBLEM

We compared the three classes of machine learning techniques to analyze gene expression data for cancer classification. All three methods were able to classify all training examples correctly. For test samples, neural trees and RBF networks showed almost perfect prediction accuracy. Only one test sample was misclassified. The misclassified sample is the 66th sample. The best Bayesian network classifier misclassified two test samples, the 66th and 67th samples. Table 5 summarizes the performances of three machine learning techniques.

Table 5. Comparison of the performance of three machine learning techniques. This table shows the best prediction performance of each method. The 66th sample is misclassified in all three methods.

	Training error	*Test error*
Bayesian networks	0/38	2/34
Neural trees	0/38	**1/34**
RBF networks	0/38	**1/34**

Although all three methods show nearly perfect prediction accuracy, they have some different characteristics. Neural trees and RBF networks are regression models and their inputs are numerical values. They can deal with thousands of genes. Moreover, neural trees can select some genes that are important for automatic cancer classification through its evolutionary procedure. Bayesian networks are a kind of probabilistic graphical models and their outputs are the probability of ALL (or AML) given some gene expression levels. To learn a reliable Bayesian network from the small dataset, all gene expression levels were discretized and only four genes were selected.

In order to find significantly correlated genes to the cancer classification problem, several methods were tested, including *P*-metric, mutual information, and neural tree learning. The set of selected genes based on the *P*-metric value and that based on the mutual information value were similar. On the contrary, neural trees found somewhat different genes compared with the other two methods.

In addition to the selection of essential genes, the mining of the underlying relationships among genes is important in bioinformatics. Among three machine learning techniques, Bayesian network learning has the power to capture the relationships among genes in comprehensible format. Neural

tree learning seems the best in finding out a small set of interesting genes for effective classification.

Table 6 summarizes the comparative characteristics of machine learning techniques we used in the experiments.

Table 6. Comparative advantages of the three machine learning methods. It should be mentioned that this evaluation is relative and confined to the problem of cancer classification with gene expression levels we addressed.

	Bayesian networks	*Neural trees*	*RBF networks*
No. of manageable genes	Small	Large	Large
Learning time	Short	Long	Short
Classification performance	Good	Excellent	Excellent
Finding significant genes	No	Yes	No
Finding probabilistic relationships	Yes	No	No

CONCLUSION AND FUTURE WORK

In this paper, we applied three machine learning techniques, i.e., Bayesian networks, neural trees, and radial basis function (RBF) networks, to the cancer classification problem. We analyze their performance and characteristics in three different aspects.

The first criterion is the prediction accuracy. The number of training samples in the CAMDA Data Set 2 is only 38, while the number of genes is 7,129. So, the data sparseness problem is inevitable. According to the characteristics of each learning method, all or some gene expression levels were used as inputs. It is interesting to see that in spite of data sparseness, we can get satisfactory prediction accuracy. All three methods classify all training samples correctly and misclassify only one or two test samples. The only misclassified sample in all the methods was the 66^{th} sample and this sample is thought to have some different characteristics from other samples.

The second issue is the selection of significantly correlated genes to the classification of cancer. This point of view is of course closely related to the prediction accuracy. To resolve the problem of data sparseness, the selection of some appropriate genes is necessary. We applied the *P*-metric value [Slonim et al., 2000], mutual information value [Cover and Thomas, 1991], and neural tree learning [Zhang and Cho, 2001] to select genes significantly correlated to cancer classification. Neural trees can automatically select appropriate genes in their learning procedure and genes found by neural tree learning are somewhat different from genes selected by the other two methods. It is interesting that there are some different subsets of genes with which the classification of leukemia is possible. “Zyxin” is selected by all

three methods and thought to play a key role in the classification of acute leukemias.

The last point of view is capturing the relationships among gene expression levels. Bayesian networks have the ability to capture probabilistic relationships among gene expression levels in human comprehensible format. These probabilistic relationships are represented by edges in Bayesian networks and can be interpreted as the indications of possible causal relationships among gene expressions. Of course, the real causal relationships among gene expressions can only be revealed by further biological experiments and analyses. However, the learning of Bayesian networks may give some draft view of the possible causal relationships among genes and help the design of further biological experiments. Neural trees can also represent some functional relationships among gene expression levels but the relationships are hard to understand and hardly represent causal relationships.

As a conclusion, neural trees and RBF networks are very good classifiers. They show the excellent prediction accuracy despite the data sparseness inherent in cancer classification with gene expression levels. Bayesian network learning is a very good approach for the analysis of gene expression data because it is able to reveal probabilistic relationships among gene expression levels.

In our experiments, gene expression levels were discretized for Bayesian network learning. The applied method is very simple and more sophisticated discretization by human experts or combining discretization into the Bayesian network learning process [Friedman and Goldszmidt, 1996] will improve the performance of Bayesian networks. Data sparseness makes the construction of a reliable Bayesian network difficult. Because of the small sample size, only four genes were used to learn the Bayesian network. In the case of concentrating only on the qualitative part (structure) of the Bayesian network, the statistical methods such as bootstrap can be useful to induce a better network structure from data ([Friedman et al., 1999] and [Friedman et al., 2000]). Methods for learning the more reliable quantitative part (local probability distributions) of Bayesian networks from such a small dataset should be studied further.

ACKNOWLEDGMENTS

This work was supported in part by BK21-IT Program and the Brain Science and Engineering Program.

REFERENCES

Chickering, D.M., Learning Bayesian networks is NP-complete, In D. Fisher and H.-J. Lenz, editors, *Learning from Data: Artificial Intelligence and Statistics V*, pp. 121-130, Springer-Verlag, 1996.

Cover, T.M. and Thomas, J.A., *Elements of Information Theory*, John Wiley & Sons, 1991.

Dougherty, J., Kohavi, R., and Sahami, M., Supervised and unsupervised discretization of continuous features, In *Proceedings of the 12th International Conference on Machine Learning (ICML 1995)*, pp. 194-202, 1995.

Friedman, N. and Goldszmidt, M., Discretizing continuous attributes while learning Bayesian networks, In *Proceedings of the 13th International Conference on Machine Learning (ICML 1996)*, pp. 157-165, 1996.

Friedman, N. and Goldszmidt, M., Learning Bayesian networks with local structure, In M.I. Jordan, editor, *Learning in Graphical Models*, pp. 421-459, MIT Press, 1999.

Friedman, N., Goldszmidt, M., and Wyner, A., Data analysis with Bayesian networks: a bootstrap approach, In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI 1999)*, pp. 196-205, 1999.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D., Using Bayesian networks to analyze expression data, In *Proceedings of the 4th Annual International Conference on Computational Molecular Biology* (*RECOMB 2000*), pp. 127-135, 2000.

Golub, T.R. et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, *Science*, vol. 286, no. 5439, pp. 531-537, 1999.

Haykin, S., *Neural Networks: A Comprehensive Foundation*, 2nd edition, Prentice Hall, 1999.

Heckerman, D., Geiger, D., and Chickering, D.M., Learning Bayesian networks: the combination of knowledge and statistical data, *Machine Learning*, vol. 20, no. 3, pp. 197-243, 1995.

Heckerman, D., A tutorial on learning with Bayesian networks, Technical Report MSR-TR-95-06, Microsoft Research, 1996.

Jensen, F.V., *An Introduction to Bayesian Networks*, University College London Press, 1996.

Park, S.-W. and Zhang, B.-T., Learning constructive RBF networks by active data selection, In *Proceedings of the 7th International Conference on Neural Information Processing* (*ICONIP 2000*), pp. 1411-1415, 2000.

Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.

Platt, J., A resource-allocating network for function interpolation, *Neural Computation*, vol. 3, no. 2, pp. 213-225, 1991.

Slonim, D.K., Tamayo, P., Mesirov, J.P., Golub, T.R., and Lander, E.S., Class prediction and discovery using gene expression data, In *Proceedings of the 4th Annual International Conference on Computational Molecular Biology* (*RECOMB 2000*), pp. 263-272, 2000.

Zhang, B.-T., Effects of Occam's razor in evolving sigma-pi neural nets, In Y. Davidor, H.-P. Schwefel, and R., Männer, editors, *Lecture Notes in Computer Science*, vol. 866, pp. 462-471, Springer-Verlag, 1994.

Zhang, B.-T., Ohm, P., and Mühlenbein, H., Evolutionary induction of sparse neural trees, *Evolutionary Computation*, vol. 5, no. 2, pp. 213-236, 1997.

Zhang, B.-T. and Cho, D.-Y., System identification using evolutionary Markov chain Monte Carlo, *Journal of Systems Architecture*, vol. 47, no. 7, pp. 587-599, 2001.