2006 International Joint Conference on Neural Networks
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Applying RBF Neural Networks to Cancer Classification Based on Gene Expressions

Feng Chu and Lipo Wang
School of Electrical and Electronic Engineering
Nanyang Technological University
Block S1, Nanyang Avenue, Singapore 639798
Email: elpwang@ntu.edu.sg

***Abstract*— Accurate classification of cancers based on microarray gene expressions is very important for doctors to choose a proper treatment. In this paper, we apply a novel radial basis function (RBF) neural network that allows for large overlaps among the hidden kernels of the same class to this problem. We tested our RBF network in three data sets, i.e., the lymphoma data set, the small round blue cell tumors (SRBCT) data set, and the ovarian cancer data set. The results in all the three data sets show that our RBF network is able to achieve 100% accuracy with much fewer genes than the previously published methods did.**

I. Introduction

DNA microarrays enable researchers to monitor the expression levels of thousands of genes simultaneously [1]. With the help of gene expressions, heterogeneous cancers can be classified into appropriate subtypes [2], [3], [4]. Recently, different kinds of machine learning and statistical methods, e.g., [5], [6], [7], [8], have been used to classify cancers using microarray gene expression data.

To evaluate the effectiveness of these cancer classification methods, two criteria may be used, i.e., the classification accuracy and the number of genes used by the classifier. For a cancer classifier, the fewer the genes used, the lower the computational burden. In addition, a reduced number of genes can significantly increase the classification accuracy, because of the reduction or the absence of irrelevant genes acting as "noise" for the classifier. Perhaps more importantly, once a smaller subset of genes are identified as relevant to a particular cancer, it helps biomedical researchers focus on these genes that contribute to the development of the cancer. Therefore, finding out small gene subsets that can ensure highly reliable classification becomes a problem of both theoretical and practical importance.

For the lymphoma data set [2], Tibshirani *et al.* successfully classified the three lymphoma sub-types with only 48 genes from the original 4026 genes, by using a statistical method named nearest shrunken centroids with an accuracy of 100% [9]. For the SRBCT data [5], Khan *et al.* precisely classified the small round blue cell tumors (SRBCTs) of childhood with 96 genes by using an artificial neural network with an accuracy of 100% [5]. Tibshirani *et al.* also applied nearest shrunken centriods to the SRBCT data set and obtained 100% accuracy with 43 genes [8]. Deutsh reduced the number of genes required to correctly classify the four cancer sub-types in the SRBCT data set to 12 genes [10]. Recently, Lee and Lee also obtained 100% accuracy in this data set with a support vector machine (SVM) classifier using at least 20 genes [11]. For the ovarian data [12], Schaner *et al.* reported that they 100% correctly classified breast cancer and ovarian cancer with 61 genes by using the nearest shrunken centroids [12].

Here, we report an approach based on a novel radial basis function (RBF) neural network that successfully classified the lymphoma data set [2] with 100% accuracy using only 9 genes. This approach also obtained 100% accuracy in the SRBCT data set [5] and the ovarian data [12] with only 8 genes and 4 genes, respectively. Our method includes two steps. In the first step, we select some genes with the greatest discriminative ability in the training data. In the second step, we use the selected genes to train our RBF neural network and subsequently use the trained network to classify the testing data.

The paper is organized as follows. In Section II, we introduce a t-test-based gene discriminative ability ranking approach. In Section III, we decribe a novel RBF neural network that we have proposed recently [16]. We applied our RBF neural network to the lymphoma, the SRBCT, and the Ovarian gene expression data sets in Section IV. In the final section, we compared our approach with previously proposed ones and draw our conclusion.

II. Gene Discriminative Ability Ranking

Typical gene expression data sets contain the expression profiles of a large number of genes, usually from several thousands to tens of thousands. However, the discriminative ability varies greatly from gene to gene. In this paper, we use a t-test scoring method (t-score) [13], [14] to measure the discriminative ability of genes. The higher the gene's t-score (TS), the higher its discriminative ability. The TS of gene i is defined as follows [13]:

$$TS_i = max\{|\frac{\overline{x}_{ik} - \overline{x}_i}{m_k s_i}|, k = 1, 2, ...K\} \quad (1)$$

$$\overline{x}_{ik} = \sum_{j \in C_k} \overline{x}_{ij}/n_k \quad (2)$$

$$\overline{x}_i = \sum_{j=1}^{n} x_{ij}/n \quad (3)$$

0-7803-9490-9/06/$20.00/©2006 IEEE

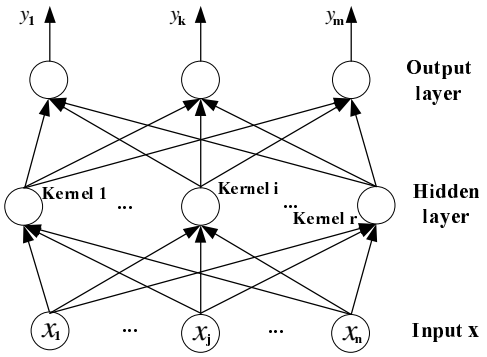

Fig. 1. The structure of an RBF neural network.

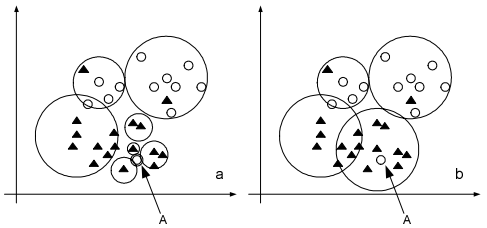

Fig. 2. A description of our RBF algorithm.

$$s_i^2 = \frac{1}{n-K}\sum_k \sum_{j\in C_k} (x_{ij} - \overline{x}_{ik})^2 \quad (4)$$

$$m_k = \sqrt{1/n_k - 1/n} \quad (5)$$

There are K classes. $max\{y_k, k = 1, 2, ...K\}$ is the maximum of all y_k. C_k refers to class k that includes n_k samples. x_{ij} is the expression value of gene i in sample j. $\overline{x}_{ik}$ is the mean expression value in class k for gene i. n is the total number of samples. $\overline{x}_i$ is the general mean expression value for gene i. s_i is the pooled within-class standard deviation for gene i. Actually, the TS used here is a t-statistic between a specific class and the overall centroid of all the classes [13].

III. RBF Neural Network

An RBF neural network [15] has three layers as shown in Fig.1. The first layer is an input layer; the second layer is a hidden layer that includes some radial basis functions, also known as hidden kernels; and the third layer is the output layer. An RBF neural network can be considered as a mapping of input domain X onto the output domain Y.

$$y_m(\vec{x}) = \sum_{i=1}^{N} w_{mi} G(\|\vec{x} - \vec{t_i}\|) + b_m,$$
$$i = 1, 2, ..., N; m = 1, 2, ..., M \quad (6)$$

Here $\|\cdot\|$ stands for the Euclidean norm. M is the number of outputs. N is the number of hidden kernels. $y_m(\vec{x})$ is output m corresponding to the input $\vec{x}$. $\vec{t_i}$ is the center of kernel i. w_{mi} is the weight between kernel i and output m. b_m is the bias on output m. $G(\|\vec{x}-\vec{t_i}\|)$ is the kernel function. The most commonly used kernel function for RBF neural networks is Gaussian kernel function as follows:

$$G(\|\vec{x} - \vec{t_i}\|) = exp(-\frac{\|\vec{x} - \vec{t_i}\|^2}{2{\sigma_i}^2}) \quad (7)$$

where σ_i is the radius of the kernel i. The main steps to construct an RBF neural network include: (a) determining the positions of all the kernels $\vec{t_i}$, (b) determining the radius of each kernel, and (c) calculating the weights between the kernels and the output nodes.

In this paper, we use a novel RBF neural network proposed by Fu and Wang [16], which allows for large overlaps of hidden kernels of the same class. The following steps describe the algorithm to generate the RBF neural network. First of all, we divide all the data into two parts, i.e., the data for training (V) and the data for testing (V_t). After that, we make a copy of V, named as V_c that is used for selecting centers of hidden kernels. Then we randomly select one pattern x_k in V_c as the center of a kernel. We search for all the patterns within a δ-neighborhood of x_k, i.e., all the patterns whose distances from x_k are less than δ. These patterns form a kernel. Subsequently, we check the purity of this kernel, that is, the ratio between the number of patterns of the same class in the kernel and the total number of patterns in this kernel. If the purity of a kernel is larger than a pre-defined threshold θ, this kernel is a qualified kernel. Otherwise, we shrink the kernel gradually by reducing δ step by step until the purity becomes larger than θ, i.e. the kernel becomes a qualified one. Once a qualified kernel is generated, all the patterns of this kernel are moved out from V. We keep on generating kernels until all the patterns are moved out from V. Because all the centers are selected from V_c rather than V, our algorithm therefore allows large overlaps among kernels of the same class. The rationale for this modification is as follows. Small overlaps among kernels improve generalization over cases without overlaps; however accuracy will suffer if the overlaps (among kernels of different classes) are too large. We note that overlaps among kernels of the same class do not decrease accuracy no matter how large they are, and at the same time, they help increase kernel size and robustness of the RBF network [16].

To start shrinking a kernel from a proper value, we set the standard deviation of all the training patterns as the initial value of δ (δ_0). After obtaining all the kernels, we set the centroid of each kernel as its center and the standard deviation of each kernel as its radius. Finally, the weights between hidden kernels and outputs are obtained with the linear least square method [17].

Compared to the RBF neural network proposed in [18], our approach allows for large overlaps among kernels of the same class. Such overlap helps the network reject "noise" and therefore improves the classification accuracy, as depicted in Fig. 2. In Fig.2(a), pattern A is surrounded by patterns belonging to another class (In such a case, A is very likely to be a "noise"). If we form kernels with the traditional method [18], pattern A will "smash" a large kernel into some smaller

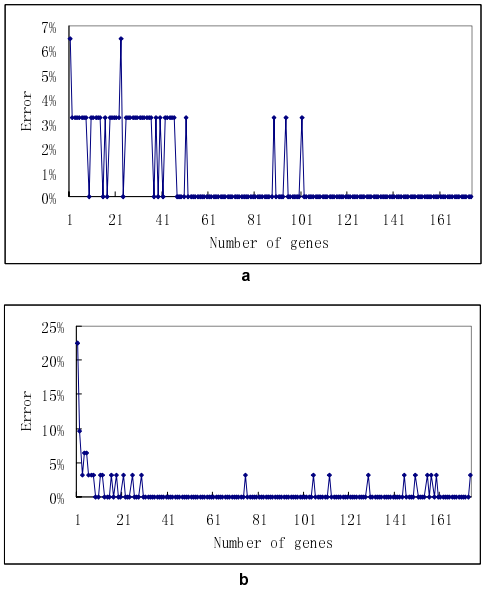

Fig. 3. The (a) training and (b) testing results for the lymphoma data set.

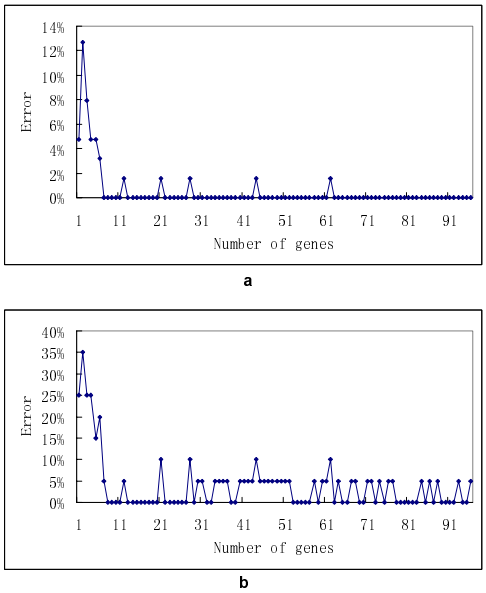

Fig. 4. The (a) training and (b) testing results for the SRBCT data set.

ones (Fig.2). However, if large overlap among kernels of the same class is allowed, a large kernel will be formed (Fig.2(b)). This large kernel can reject the influence of pattern A (noise). Furthermore, it can reduce the number of hidden kernels and improve the efficiency of the construction of the network [16].

IV. Results

A. Lymphoma Data

The lymphoma data set (http://llmpp.nih.gov/lymphoma) [2] contains 42 samples derived from diffuse large B-cell lymphoma (DLBCL), 9 samples from follicular lymphoma (FL), 11 samples from chronic lymphocytic leukaemia (CLL). The entire data set includes the expression data of 4026 genes. In this data set, a small part of data is missing. A k-nearest neighbor algorithm was applied to fill those missing values [19].

At first, we randomly divided the 62 samples into 2 parts, 31 samples for training, 31 samples for testing. we ranked the entire 4026 genes according to their TSs with the 31 training samples. Then we picked out the 174 genes with the highest TSs (Table 1). We subsequently input the selected 174 genes one by one to the network according to their TS ranks starting with the gene ranked 1 in Table 1. That is, we first used only a single gene that is ranked 1 as the input to the network. We trained the network with the training data, and subsequently tested the network with the test data. We repeated this process with the first 2 genes in Table 1, then the first 3 genes, and so on. The training and testing results are shown in Fig.3. We found that the RBF network performed very well: its training error and testing error both decreased to 0 with only the first 9 genes in Table 1.

B. SRBCT Data

The SRBCT data (http://research.nhgri.nih.gov/ microarray/Supplement/) [5] contains the expression data of 2308 genes. There are totally 63 training samples and 25 testing samples provided, 5 of the testing samples are not SRBCTs. The 63 training samples contain 23 Ewing family of tumors (EWS), 20 rhabdomyosarcoma (RMS), 12 neuroblastoma (NB) and 8 Burkitt lymphomas (BL). And the 20 SRBCT testing samples contains 6 EWS, 5 RMS, 6 NB and 3 BL.

We followed the same procedure as what we did in the lymphoma data set. We firstly ranked the entire 2308 genes according to their TSs [13][14] with the 63 training samples. Then we picked out the 96 genes with the highest TSs. We input the selected 96 genes one by one to the RBF neural network according to their TSs in the decreasing order. Fig.4 shows the training and the testing errors happened during classification. Both the training error and the testing error decreases to 0 when the top 8 genes are input into the RBF network.

C. Ovarian Data

The ovarian data (http://genome-www.stanford.edu/ovarian_cancer/) [12] contains 125 samples, including 68 samples derived from breast cancer and 57 samples derived from ovarian cancer. The entire data set includes the expression data of 3363 genes.

Similarly, we first randomly divided the data into 2 parts, 75 samples for training, 50 samples for testing. We ranked the entire 3363 genes according to their TSs with the 75 training samples. Then we picked out the 100 genes with the highest TSs. We subsequently input the selected 100 genes one by one to the network according to their TSs in the decreasing order. Fig.5 shows the training and the testing results. Form these

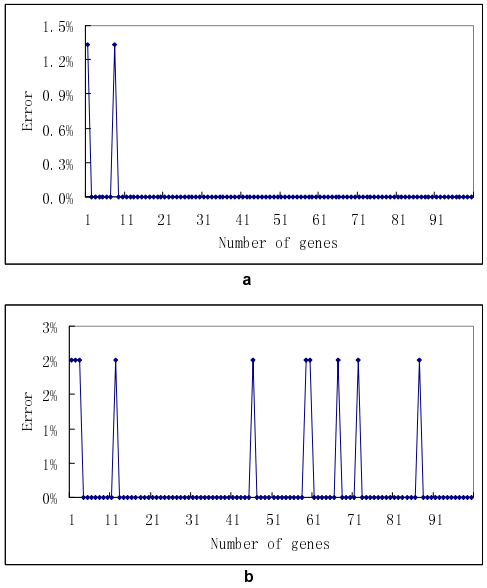

Fig. 5. The (a) training and (b) testing results for the ovarian data set.

results, we found that our RBF network also performed very well: both the training error and the testing error decreased to 0 when only the top 4 genes are input inot the RBF network.

V. Discussion

For the lymphoma data, to our knowledge, the best published result was obtained by the nearest shrunken centroids [9], which used 48 genes to 100% correctly classify the three types of lymphoma. Compared to the nearest shrunken centroids [9], our RBF neural network used only 9 genes to obtain the same accuracy.

For the SRBCT data, our RBF neural network also required much fewer genes to achieve 100% accuracy than the previously published results [5], [8], [10], [11] did. A comparison is given in table 2.

For the ovarian data, our RBF neural network used only 4 genes to obtain 100% accuracy. In contrast, [12] correctly classified this data with a minimal set of 61 genes. The method they used is the nearest shrunken centroids [8].

In view of the results of our RBF network and the comparison with other popular methods in all the three data sets, we conclude that our RBF neural network can greatly reduce the number of genes (and hence greatly reduce the gene redundancy) required for accurate classification of cancers using microarray gene expressions.

References

[1] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," *Science*, vol. 270, pp. 467-470, 1995.

[2] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, *et al.*, "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, pp. 503-511, 2000.

[3] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, *et al.*, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science,* vol. 286, pp. 531-537, 1999.

[4] X. Ma, R. Salunga, J. T. Tuggle, J. Gaudet, E. Enright, P. McQuary, T. Payette, M. Pistone, K. Stecker, B. M. Zhang, *et al.*, "Gene expression profiles of human breast cancer progression," *Proc. Natl. Acad. Sci. USA*, vol. 100, pp. 5974-5979, 2003.

[5] J. M. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, *et al.*, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature Medicine*, vol. 7, pp. 673-679, 2001.

[6] M. P. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Jr. Ares, D. Haussler, "Knowledge-based analysis of microarray gene expression data by using support vector machines," *Proc. Natl Acad. Sci. USA*, vol. 97, pp. 262-267, 2000.

[7] A. B. Olshen, and A. N. Jain, "Deriving quantitative conclusions from microarray expression data," *Bioinformatics*, vol. 18, pp. 961-970, 2002.

[8] R. Tibshirani, T. Hastie, B. Narashiman, and G. Chu, "Diagnosis of multiple cancer types by shrunken centroids of gene expression," *Proc. Natl. Acad. Sci. USA*, vol. 99, pp. 6567-6572, 2002.

[9] R. Tibshirani, T. Hastie, B. Narasimhan, G. Chu, "Class predicition by nearest shrunken centroids with applications to DNA microarrays," *Statistical Science*, vol. 18, pp. 104-117, 2003.

[10] J. M. Deutsch, "Evolutionary algorithms for finding optimal gene sets in microarray prediction," *Bioinformatics*, vol. 19, pp. 45-52, 2003.

[11] Y. Lee, and C. K. Lee, "Classification of multiple cancer types by mulitcategory support vector machines using gene expression data," *Bioinformatics*, vol. 19, pp. 1132-1139, 2003.

[12] M. E. Schaner, D. T. Ross, G. Ciaravino, T. Sorlie, O. Troyanskaya, M. Diehn, Y. C. Wang, G. E. Duran, T. L. Sikic, S. Caldeira, "Gene expression patterns in ovarian carcinomas," *Molecular Biology of the Cell*, vol. 14, pp. 4376-4386, 2003.

[13] J. Devore, and R. Peck, *Statistics: the Exploration and Analysis of Data. 3rd edition*, Pacific Grove, CA: Duxbury Press, 1997.

[14] V. G. Tusher, R. Tibshirani, and G. Chu, "Significance analysis of microarrays applied to the ionizing radiation response," *Proc. Natl. Acad. Sci. USA*, vol. 98, pp. 5116-5121, 2001.

[15] S. Haykin, *Neural networks: a comprehensive foundation 2nd Ed.* N.J.: Prentice-Hall, 1999.

[16] X. Fu, and L. Wang, "Data dimensionality reduction with application to simplifying RBF network structure and improving classification performance," *IEEE Trans. Systems, Man, Cybernetics-Part B: Cybernetics*, vol. 33, pp. 399-409, 2003.

[17] C. M. Bishop, *Neural Networks for Pattern Recognition.* New York: Oxford University Press, 1995.

[18] A. Roy, S. Govil, and R. Miranda, "A Neural-Network Learning Theory and a Ploynomial Time RBF Algorithm," *IEEE Trans. Neural Networks*, vol. 8, pp. 1301-1313, 1997.

[19] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, R. B. Altman, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, pp. 520-525, 2001.

TABLE I

LYMPHOMA GENE IMPORTANCE RANKING : 174 GENES WITH THE HIGHEST TSS, IN THE ORDER OF DECREASING TSS (GENE ID IS DEFINED IN [2])

Rank	Gene ID	Gene Description
1	GENE2307X	(CD23A=low affinity II receptor for Fc fragment of IgE; Clone=1352822)
2	GENE3320X	(Similar to HuEMAP=homolog of echinoderm microtubule associated protein EMAP; Clone=1354294)
3	GENE708X	*Ki67 (long type); Clone=100
4	GENE2393X	*MDA-7=melanoma differentiation-associated 7=anti-proliferative; Clone=267158
5	GENE1622X	*CD63 antigen (melanoma 1 antigen); Clone=769861
6	GENE1641X	*Fibronectin 1; Clone=139009
7	GENE2391X	(Unknown; Clone=1340277)
8	GENE1636X	*Fibronectin 1; Clone=139009
9	GENE1644X	(cathepsin L; Clone=345538)
10	GENE1610X	*Mig=Humig=chemokine targeting T cells; Clone=8
11	GENE707X	(Topoisomerase II alpha (170kD); Clone=195630)
12	GENE689X	*lamin B1; Clone=1357243
13	GENE695X	*mitotic feedback control protein Madp2 homolog; Clone=814701
14	GENE1647X	*cathepsin B; Clone=261517
15	GENE537X	(B-actin,1099-1372; Clone=143)
...	...	...
165	GENE1539X	*lysophospholipase homolog (HU-K5); Clone=347403
166	GENE2385X	*Unknown UG Hs.124382 ESTs; Clone=1356466
167	GENE719X	(Myt1 kinase; Clone=739511)
168	GENE2415X	(Unknown; Clone=1289937)
169	GENE527X	*glutathione-S-transferase homolog; Clone=1355339
170	GENE1598X	*Similar to ferritin H chain; Clone=1306027
171	GENE1192X	*Interferon-induced guanylate-binding protein 2; Clone=545038
172	GENE731X	*Chromatin assembly factor-I p150; Clone=1334875
173	GENE769X	*14-3-3 epsilon; Clone=266106
174	GENE724X	(Hyaluronan-mediated motility receptor (RHAMM); Clone=756037)

TABLE II

COMPARISONS OF RESULTS ALL WITH 100% ACCURACY FOR THE SRBCT DATA

Method	Number of genes required
MLP neural network [5]	96
Nearest shrunken centroids [8]	43
SVM [11]	20
Evolutionary algorithm [10]	12
Our RBF neural network	8