Microarray classification from several two-gene expression comparisons

Donald Geman, Bahman Afsari, Aik Choon Tan, and Daniel Q. Naiman

***Abstract*— We describe our contribution to the ICMLA2008 "Automated Micro-Array Classification Challenge". The design of our classifier is motivated by the special scenario encountered in molecular cancer classification based on the mRNA concentrations provided by gene microarray data. Our classifier is rank-based; it only depends on expression comparisons among selected pairs of genes. Such comparisons are invariant to most of the transformations involved in preprocessing and normalization. Every pair of genes determines a binary classifier - choose the class for which the observed ordering is most likely. Pairs are scored by maximizing accuracy. In our *k-TSP* (*k*-disjoint *T*op *S*coring *P*airs) classifier, *k* disjoint pairs of genes are learned from training data; the discriminant function is simply the difference in the number of votes for the two classes. This rule involves exactly 2*k* genes, is readily interpretable, and provides some state-of-the-art results in cancer diagnosis and prognosis for small values of *k*, even *k*=1.**

Index Terms— **Molecular classification, gene expression, cancer diagnosis, rank-based, maximum likelihood.**

I. Introduction

High-throughput microarray technology provides a powerful tool in biomedical research. Specifically, DNA microarray profiling technology has shown to be useful in disease diagnosis and prognosis, as well as subtype identification [1-5]. The development of innovative computational algorithms, especially statistical and machine learning approaches, has contributed to interpreting gene expression data. In particular, extracting accurate and simple decision rules from such microarray data for cancer classification and prediction is of great interest in molecular biology and medicine. However, one serious limitation of current approaches, especially for making a transition from fundamental research to eventual clinical use, is the black-box nature of the decision rules resulting from standard machine learning methods. Classification is a mystery for non-specialists. The work presented here is motivated by this interpretability dilemma.

We describe a classifier involving *2k* genes arranged in *k* pairs; the decision rule is based on voting among the *k* two-gene expression comparisons. This *k-TSP* (*k*-disjoint *T*op *S*coring *P*airs) classifier for labeling gene expression data was presented in [6]. When *k=1*, this algorithm, referred to simply as *TSP* [7], necessarily selects a single pair of genes. The main differences between the algorithm presented here and the one in [6] are i) here *k* is fixed rather than regarded as a parameter to be estimated during learning; and ii) the *2k* genes are restricted to a list of differentially expressed genes in order to expedite training We formulate the gene expression classification problem in section 2 and describe the technical details of the *k-TSP* algorithm in section 3. We then evaluate its performance in section 4 and draw some conclusions in section 5.

II. TRAINING DATA

A gene expression profile consists of P genes $\{1,\ldots,P\}$ and N profiles or arrays, $\mathbf{x}_1,\ldots,\mathbf{x}_N$,. These data can be represented as a matrix of dimension $P \times N$ in which the expression value of the i-th gene, $i \in \{1,\ldots,P\}$, from the n-th sample is denoted by $x_{i,n}$. The column vector $\mathbf{x}_n = (x_{1,n},\ldots,x_{P,n})$ represents the P expression values for the n-th sample. Let $(y_1,\ldots,y_N)$ be the vector of class labels for the N samples, where $y_n \in \mathbf{C}$, the set of possible class labels. For simplicity, we assume a binary classification problem, $M = 2$; for example, $Y=1$ refers to normal tissues and $Y=-1$ to cancer tissues. The labeled training set is then $S = \{(\mathbf{x}_1, y_1),\ldots,(\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_n$ is the n-th column vector of the matrix of gene expression profiles. As usual, the expression profile and its class label are regarded as random variables, denoted by $\mathbf{X}$ and Y respectively, and the elements of S is assumed to represent independent and identically distributed samples from the underlying probability distribution of $(\mathbf{X}, Y)$.

III. K-TSP CLASSIFIER

The *k-TSP* classifiers are *rank-based*, meaning that the decision rules only depend on the relative ordering of the gene expression values within each profile [6]. The

Manuscript received June 15, 2008.

Donald Geman is with the Department of Applied Mathematics and Statistics, Center for Imaging Sciences and the Institute for Computational Medicine, Johns Hopkins University, Baltimore MD. 21234 USA (corresponding author, phone: 410-516-7678; fax: 410-516-4594; e-mail: geman@ jhu.edu).

Bahman Afsari is with the Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore MD. 21234 USA (e-mail: bafsari1@jhu.edu).

Aik Choon Tan is with the Sidney Kimmel Comprehensive Cancer Center at Johns Hopkins University School of Medicine, Baltimore, MD. 21231 USA (e-mail: actan@jhu.edu).

Daniel Q. Naiman is with the Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore MD. 21234 USA (e-mail: daniel.naiman@ jhu.edu).

expression values of the P genes are ordered (most highly expressed, second most highly expressed, etc.) within each fixed profile. Let $R_{i,n}$ denote the rank of i-th gene in the n-th array (profile). Replacing the expression values $x_{i,n}$ by their ranks $R_{i,n}$ results in a new data matrix **R** in which each column is a permutation of $\{1,\ldots,P\}$.

A. *Learning the k-TSP classifier.*

The *k-TSP* classifier has been described previously. In essence, information contained in the **R** matrix will be exploited by focusing on "marker gene pairs" (i, j) for which there is a significant difference in the probability of the event $\{ R_i < R_j \}$ across the N samples from class $Y=1$ to $Y=-1$. Notice that the event $\{ R_i < R_j \}$ is equivalent to the event $\{ X_i < X_j \}$; that is, the rank of gene i is less than the rank of gene j if and only if gene i is expressed less than gene j. Here, the quantities of interest are $p_{ij}(m) = \text{Prob}(R_i < R_j \mid Y=m)$, $m = \{1,-1\}$, i.e., the probabilities of observing $R_i < R_j$ in each class. These probabilities are estimated by the relative frequencies of occurrences of $R_i < R_j$ within profiles and over samples. Let Δ_{ij} denote the "score" of gene pair (i, j), where $\Delta_{ij} = | p_{ij}(1) - p_{ij}(-1)|$. We compute the score Δ_{ij} for every pair of genes $i, j \in \{1,\ldots,P\}$, $i \neq j$. Obviously, pairs of genes with high scores are viewed as most informative for classification. It is easy to show that maximizing the score is equivalent to minimizing the sum of the two error rates for the classifier which chooses the class for which the observed ordering between the expressions of genes i and j is the most likely. The *k-TSP* classifier utilizes the k *disjoint* pairs of genes which achieve the k best scores.

The selection process is the following. First, a ranked list of all pairs is compiled based on the score. In order to distinguish among pairs achieving the same score, we use a secondary score based on the rank differences in each sample in each class. The motivation behind using the rank score to break ties is that it incorporates a measure of the magnitude to which inversions of gene expression levels occur from one class to the other within a pair of genes. For each top-scoring gene pair (i, j), we compute the "average rank difference" γ_{ij} in class m defined as:

$$\gamma_{ij}(m) = \frac{\sum_{n \in C_m} (R_{i,n} - R_{j,n})}{|C_m|}, \qquad m = \{1,-1\} \tag{1}$$

where C_m denotes the set of samples in class m and $|C_m|$ is the number of samples in class m. The "rank score" of gene pair (i,j) is then defined to be $\Gamma_{ij} = | \gamma_{ij}(1) - \gamma_{ij}(-1)|$. For any given score, we order the pairs achieving that score using the rank score.

Finally, once every pair of genes is assigned a unique rank, we recursively choose the top disjoint k pairs. The top-ranked pair is automatically selected. Then we choose the highest ranking pair with no gene in common with the top pair, and so forth, until k pairs of genes are selected.

B. *Prediction with the k-TSP classifier.*

Each of the selected pairs (i, j) defines a classifier, namely the maximum likelihood classifier based on the observed ordering of X_i and X_j. Suppose $p_{ij}(1) > p_{ij}(-1)$. Then, given a profile **x**, the classifier $f_{ij}(\mathbf{x})$ based on this pair (i,j) is

$$f_{ij}(\mathbf{x}) = \begin{cases} 1, & X_i < X_j, \\ -1, & otherwise. \end{cases} \tag{2}$$

If, on the other hand, if $p_{ij}(-1) \geq p_{ij}(1)$, then the decision rule is reversed.

For k gene pairs $\{(i_1,j_1),\ldots, (i_k,j_k))$, we define the discriminant function, $f(\mathbf{x}) = \sum_{l=1}^{k} f_{i_l j_l}(\mathbf{x})$. The *k-TSP* classifier predicts the new profile **x** as class 1 if $f(\mathbf{x})$ is positive and class -1 if $f(\mathbf{x})$ is negative.

C. *Implementation of the k-TSP classifier*

The fully automated *k-TSP* classifier is implemented in Matlab using an interface suggested by the ICMLA2008 rules, with two key components: a module for building the classifier using training data, and a module for using an existing classifier to make predictions. The reader is warned that in order to conform to the requirements of the competition, expression data for given samples are represented in this section as row vectors rather than as column vectors.

Training function is invoked as follows:

[model, index] = train(z, y)

where z is a matrix whose rows correspond to training samples and its columns correspond to genes, i.e. z_{ij} is the expression level of the j-th gene in the micro-array corresponding to the i-th sample in the training set. The other argument to the function is a column vector, y, whose i-th component y_i is the label ±1 of the i-th sample in z.

The "model" portion of the output is a $2 \times k$ matrix, each of whose l-th columns consists of the ordered pair of gene indices (i_l,j_l) defining one of the top-scoring pairs. The variable "index" is simply a list of these $2k$ genes, so that the model and index components have the same elements. The

redundant index component is included simply to satisfy the requirements of the competition and is not needed for prediction.

To use the classifier for prediction, one can call the following prediction function using:

y = predict(model, *z*)

where "model" variable comes from the output of the training function just describe, and *z* is a matrix whose rows have the same dimension of the rows of the training matrix *z*, and correspond to samples to be classified. Each pair of genes in the classifier *votes* for a class label, and hence contributes a value of either -1 or 1, and these *k* votes are summed to produce an output score. Thus the output *y* is a column giving the voting score for each row of *z* to be classified.

IV. RESULTS

To evaluate the performance of *k-TSP* classifier, we performed leave-one-out cross-validation (LOOCV) on two benchmark datasets provided by the ICMLA 2008 Automated Microarray Classification Challenge organizers. The first dataset is generated by Golub et al [2] for distinguishing acute myeloid leukemia (AML) from acute lymphoblastic leukemia (ALL) (N = 72, P = 7,129). The second dataset is generated by Alon et al [1] and aims to discriminate colon tumors from normal tissues (N = 62, P = 2,000). To increase the computational efficiency, we performed gene filtering to select top 100 and 200 genes using Wilcoxon-Mann-Whitney (WMW) test. It is important to note that the WMW test is based on the rank data in the **R** matrix, not the original expression values. In this way, the entire method is invariant to most of the common preprocessing and normalization methods for gene chips. We fixed k = 5, 7, 9 and 11 in this experiment. LOOCV of the experiment is tabulated in Table 1.

Table 1: Leave-one-out cross-validation accuracy on the benchmark datasets.

Data-set		***k* (# of gene pairs)**			
	# of filtered genes	**5**	**7**	**9**	**11**
Golub et al	100	95.83%	98.61%	98.61%	98.61%
	200	95.83%	98.61%	98.61%	98.61%
Alon et al	100	87.10%	88.71%	88.71%	87.10%
	200	88.71%	88.71%	88.71%	88.71%

V. CONCLUSIONS

Here, we describe the *k-TSP* algorithm and implementation of this classifier in Matlab for the ICMLA 2008 Automated Microarray Classification Challenge. Previously, we have demonstrated that *k-TSP* classifier performs as well as Predictive Analysis of Microarray and Support Vector Machine and outperforms other learning methods (decision trees, k-nearest neighbor and Naïve Bayes), over a wide variety of cancer datasets [6]. Due to the small number of genes involved in the decision, the classifier is easy to interpret and facilitates follow-up studies [3]. Moreover, we have shown that, due to normalization-invariance, this is a natural method to induce signatures and build classifiers by aggregating data across studies and platforms in order to increase sample size [8,9].

REFERENCES

[1] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," PNAS, vol. 96, pp. 6745-6750, June 8, 1999 1999.

[2] 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, C. D. Bloomfield, and E. S. Lander, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," Science, vol. 286, pp. 531-537, October 15, 1999 1999.

[3] N. D. Price, J. Trent, A. K. El-Naggar, D. Cogdell, E. Taylor, K. K. Hunt, R. E. Pollock, L. Hood, I. Shmulevich, and W. Zhang, "Highly accurate two-gene classifier for differentiating gastrointestinal stromal tumors and leiomyosarcomas," PNAS, vol. 104, pp. 3414-3419, February 27, 2007 2007.

[4] M. J. van de Vijver, Y. D. He, L. J. van 't Veer, H. Dai, A. A. M. Hart, D. W. Voskuil, G. J. Schreiber, J. L. Peterse, C. Roberts, M. J. Marton, M. Parrish, D. Atsma, A. Witteveen, A. Glas, L. Delahaye, T. van der Velde, H. Bartelink, S. Rodenhuis, E. T. Rutgers, S. H. Friend, and R. Bernards, "A Gene-Expression Signature as a Predictor of Survival in Breast Cancer," N Engl J Med, vol. 347, pp. 1999 - 2009, 2002.

[5] L. J. van 't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. M. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend, "Gene expression profiling predicts clinical outcome of breast cancer," Nature, vol. 415, pp. 530 - 536, 2002.

[6] A. C. Tan, D. Q. Naiman, L. Xu, R. L. Winslow, and D. Geman, "Simple decision rules for classifying human cancers from gene expression profiles," Bioinformatics, vol. 21, pp. 3896-3904, October 15, 2005 2005.

[7] D. Geman, C. d'Avignon, D. Q. Naiman, and R. L. Winslow, "Classifying Gene Expression Profiles from Pairwise mRNA Comparisons," Statistical Applications in Genetics and Molecular Biology, vol. 3, p. Article 19, 2004.

[8] L. Xu, A. C. Tan, D. Q. Naiman, D. Geman, and R. L. Winslow, "Robust prostate cancer marker genes emerge from direct integration of inter-study microarray data," Bioinformatics, vol. 21, pp. 3905-3911, October 15, 2005 2005.

[9] L. Xu, A. C. Tan, R. Winslow, and D. Geman, "Merging microarray data from separate breast cancer studies provides a robust prognostic test," BMC Bioinformatics, vol. 9, p. 125, 2008.