A GENERALIZED OUTPUT-CODING SCHEME WITH SVM FOR MULTICLASS MICROARRAY CLASSIFICATION

LI SHEN ENG CHONG TAN

School of Computer Engineering, Nanyang Technological University,Nanyang Avenue, Singapore, 639798, Singapore

Multiclass cancer classification based on microarray data is described. A generalized output-coding scheme combined with binary classifiers is used. Different coding strategies, decoding functions and feature selection methods are combined and validated on two cancer datasets: GCM and ALL. The effects of these different methods and their combinations are then discussed. The highest testing accuracies achieved are 78% and 100% for the two datasets respectively. The results are considered to be very good when compared with the other researchers' work.

1 Introduction

DNA microarrays can contain thousands of gene expression levels in one single experiment. Obtaining gene expression levels from tumor tissues can help us to understand the activities of genes underlying different cancers. Therefore, these expression data may also be used to identify types or subtypes of cancers.

Applying machine learning techniques to microarray data for cancer classification has been intensively researched in recent years. Most of the work is in the field of binary classification and very high accuracy can be obtained [1]. However, it is suggested by some authors that multiclass classification tasks are more difficult than binary ones [2].

In this paper, a generalized output-coding scheme has been applied to multiclass microarray classification. With this, different coding strategies and decoding functions can be put into one single framework. The validity of various combinations has been verified. Support Vector Machine (SVM) was chosen as the binary classifier, which has been successfully applied to microarray classification. It is one of the state-of-the-art machine learning techniques and has strong theoretical foundation.

Because microarray data has the characteristics that the number of genes is much larger than the number of samples, feature selection is also important before classification. Three major categories of feature selection methods have been tested: gene ranking, recursive feature elimination (RFE) and dimension reduction.

2 Methods

2.1 Output-Coding for Multiclass Classification

Assume we have a set of m microarray samples: $(\mathbf{x}_i, y_i)$, $i = 1, 2, \ldots m$, where $\mathbf{x}_i \in \Re^n$ is a vector of length n representing gene expression levels and $y_i \in \{1, 2, \ldots, k\}$ is the class label of the i th sample. The multiclass classification algorithm aims to find the mapping $\mathrm{M} : \Re^n \to \{1, 2, \ldots, k\}$ using the m training samples. Output-coding methods solve the

multiclass problem by decomposing the k -class problem into a set of l binary subproblems, training the resulting l base classifiers and then combining the l outputs to predict the class label. We have adopted the generalized scheme proposed by [3]. It begins with a given coding matrix

$$\mathbf{M} \in \{-1, 0, +1\}^{k \times l}$$

for which each row $\mathbf{r}_i$ ($i = 1, 2, \ldots k$) represents the codeword of the i th class and each column $\mathbf{s}_j$ ($j = 1, 2, \ldots l$) represents the j th base classifier. Each row $\mathbf{r}_i$ must be unique for its corresponding class. $\mathbf{M}(i, j) = 1$ or -1 means that the i th class should be considered as positive or negative for the j th base classifier, respectively. If $\mathbf{M}(i, j) = 0$, the i th class is simply ignored by the j th base classifier. Any binary classifier can be used to solve the induced two-class problem.

Now let f_s ($s = 1, 2, \ldots l$) denote the l base classification functions. Given a microarray sample $\mathbf{x}$, let $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots f_l(\mathbf{x}))$; then its class label y is predicted as

$$y = \arg\min_i d(\mathbf{r}_i, \mathbf{f}(\mathbf{x})) \tag{1}$$

where d is called the decoding function. By adopting this generalization scheme, we can combine some of the researchers' work into one single system.

There are various methods to generate coding matrices. Different coding matrices may have substantial effect on classification accuracy. Probably the simplest coding approach is to set $\mathbf{M}$ as a square matrix of size $k \times k$. Let all diagonal elements of $\mathbf{M}$ be 1 and all the other elements be -1 . Therefore, it equals the method that creates one binary problem for each of the k classes. This is called one-versus-all (OVA) approach.

Another approach, proposed by [4], is to use the binary classifier to distinguish one pair of classes at a time. Meanwhile, the other classes are simply ignored. So there are totally C_k^2 base classifiers to induce. This is called the all-pairs (AP) approach.

Error correcting output codes (ECOC) was proposed by [5]. They argue that if the minimum hamming distance between a pair of rows of the coding matrix is c , the output codes can have the ability to correct $\lfloor (c-1)/2 \rfloor$ errors of the base classifiers. Two major coding strategies called random coding and exhaustive coding are given:

- *Random coding*. Let $l = \lceil 10 \log_2(k) \rceil$ as suggested by [5]. Each element of the coding matrix is then assigned a value from $\{-1, 1\}$ uniformly at random. After the coding matrix is generated, a hill-climbing procedure given by [6] is followed. The hill-climbing method can usually improve the averaged and minimum hamming distances between pairs of rows of the coding matrix so that the classification accuracy may be improved.
- *Exhaustive coding*. Firstly let $l = 2^{k-1}$. The codeword for the first class is all $+1$. For i , $2 \le i \le k$, the codeword for the i -th class is constructed by repeating a pattern 2^{i-2} times, which is a length- 2^{k-i} block of $+1$'s followed by a length- 2^{k-i} block of -1 's. Because the first column is thus assigned $+1$ for all of its elements, it is

deleted from the coding matrix. This makes $l = 2^{k-1} - 1$. It is easy to see that the minimum hamming distance is $\left\lfloor \frac{2^{k-1}-1}{2} \right\rfloor$. The disadvantage of exhaustive coding is that l increases exponentially with k. If k is a large number, that would make the computation intractable.

Decoding functions determine how the distance between the outputs of base classifiers and codeword is calculated. One way of doing this is to count the number of positions s in which the codeword entry differs from the sign of the prediction $f_s(\mathbf{x})$. Formally the distance measure is given as

$$d_H(\mathbf{r}, \mathbf{f}(\mathbf{x})) = \sum_{s=1}^{l} \left(\frac{1 - sign(r_s f_s(\mathbf{x}))}{2} \right) \tag{2}$$

where $sign(z) = +1$ if $z > 0$, -1 if $z < 0$ and 0 if $z = 0$. r_s is the entry of codeword $\mathbf{r}$ at position s. This is called the *hamming distance* decoding. A disadvantage of this decoding function is that it totally ignores the output values of base classifiers.

A second decoding function takes into account the confidence of the predictions. It utilizes a loss function L which is algorithm-specific. The loss function calculates the "loss" of the prediction given the output values and the codeword. The loss function for SVM is defined as

$$L(y, f) = (1 - yf)_+ \tag{3}$$

where y is an entry of the codeword and f is the output of SVM. z_+ is defined as $\max(z, 0)$. The distance measure can now be written as

$$d_L(\mathbf{r}, \mathbf{f}(\mathbf{x})) = \sum_{s=1}^{l} \left(L(r_s, f_s(\mathbf{x})) \right) \tag{4}$$

This is called the *loss based* decoding.

There is another decoding function that takes account of the prediction confidence by simply calculating the inner product of the codeword and the vector of classifier outputs. This is defined as

$$d_I(\mathbf{r}, \mathbf{f}(\mathbf{x})) = -\sum_{s=1}^{l} \left(r_s f_s(\mathbf{x}) \right) \tag{5}$$

thus the name *inner product* decoding.

Finally we introduce a decoding function that is based on the probability of the prediction. Given the assumption that the base classifiers are independent, the class of which the codeword gives the maximum joint probability is the one predicted. Negative log-likelihood can be used to define the decoding function as

$$d_P(\mathbf{r}, \mathbf{p}(\mathbf{x})) = -\sum_{s=1}^{l} \frac{1 + r_s}{2} \log(p_s(\mathbf{x})) - \sum_{s=1}^{l} \frac{1 - r_s}{2} \log(1 - p_s(\mathbf{x})) \tag{6}$$

where $\mathbf{p}(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots p_l(\mathbf{x}))$. p_s is the probabilistic output of the base classifier s. A parametric model can be used to estimate the probability of SVM outputs as suggested by [7]

$$p_s(\mathbf{x}) = \frac{1}{1+\exp(A_s f_s(\mathbf{x}) + B_s)} \tag{7}$$

where $f_s(\mathbf{x})$ is the output of SVM which is trained as base classifier s. Three-fold cross-validation (CV) is used in this paper to fit A_s and B_s.

2.2 Feature Selection

There are three major categories of feature selection methods:

- *Gene Ranking*. Intuitively one would select those genes that are correlated with a class but are uncorrelated with the other classes. We choose a gene ranking method that is based on the ratio of their between-group to within-group sums of squares. For a gene j, this ratio is

$$BW(j) = \frac{\sum_i \sum_k I(y_i = k)(\bar{x}_{kj} - \bar{x}_{\cdot j})^2}{\sum_i \sum_k I(y_i = k)(x_{ij} - \bar{x}_{kj})^2} \tag{8}$$

  where $\bar{x}_{\cdot j}$ and $\bar{x}_{kj}$ denote the average expression level of gene j across all classes and across samples belonging to class k only. $I(\cdot)$ is the indicator function. The base classifiers are built using the genes with the largest BW values.
- *Recursive Feature Elimination*. RFE was first proposed by [8] to do feature selection in binary classification. The genes with the smallest corresponding weights are dropped and the process can be executed recursively. In multiclass context, the RFE is executed on each base classifier independently so that the best performance and the smallest gene subset can be obtained concurrently. Three fold CV is used to evaluate the goodness of gene subset.
- *Partial Least Squares (PLS) and Principal Components Analysis (PCA)*. Dimension reduction methods have been proposed to tackle the "curse of dimensionality" problem. It is prohibitive to use some of the statistical methods when $m < n$ because of excessive computation time. Dimension reduction is also used as the preprocessing step to make these methods feasible. PLS [9] and PCA [10] have been proven to be effective for microarray classification [11-12] and have been used in this paper.

3 Results

3.1 Datasets and Experimental Setup

We chose two multiclass microarray datasets for our experiments. The first is the *GCM* dataset published by [13]. It consists of 144 training samples and 54 testing samples of 15 common cancer classes. Each sample has 16063 gene expression levels. For simplicity,

we dropped the 8 metastatic samples from the testing dataset because they are not present in the training dataset. Therefore, 46 testing samples and 14 cancer classes are considered in this paper. The distribution of training and testing samples among the 14 classes is listed in Table 1. The second is the *ALL* dataset published by [14]. It consists of 163 training samples and 85 testing samples of 6 subtypes of acute lymphoblastic leukemia. Each sample has 12558 gene expression levels. The distribution of training and testing samples among the 6 subtypes is listed in Table 2. All data are log-transformed. All genes are normalized to have zero mean and unit standard deviation. No other preprocessing steps are applied. For GCM data, three coding strategies are used: AP, OVA, and random. We did not use exhaustive coding because l would be equal to $2^{13}-1=8191$ and this will make the computation intractable. For ALL data, AP, OVA and exhaustive coding strategies are used.

Table 1. GCM: number of samples per cancer class. BR=Breast, PR=Prostate, LU=Lung, CO=Colorectal, LY=Lymphoma, BL=Bladder, Melanoma=ME, UT=Uterus, LE=Leukemia, RE=Renal, PA=Pancreas, OV=Ovary, ML=Mesothelioma, CNS=Brain.

Cancer Class	*BR*	*PR*	*LU*	*CO*	*LY*	*BL*	*ME*	*UT*	*LE*	*RE*	*PA*	*OV*	*ML*	*CNS*
Training	8	8	8	8	16	8	8	8	24	8	8	8	8	16
Testing	3	2	3	3	6	3	2	2	6	3	3	3	3	4

Table 2. ALL: number of samples per subtype.

Subtype	*BCR-ABL*	*E2A-PBX1*	*Hyperdiploid>50*	*MLL*	*T-ALL*	*TEL-AML1*
Training	9	18	42	14	28	52
Testing	6	9	22	6	15	27

According to the suggestion of [2], 250 top genes are selected from BW ratio ranking. We also tested the data without feature selection, which is denoted as NO in the following. For RFE, the gene subset for each base classifier is determined by three fold CV. For PLS and PCA, all components that can be extracted are used. All programs are written in MATLAB codes. The software package written by Steve Gunn is used for the SVM algorithm. It is available at: http://www.kernel-machines.org/. The regularization parameter for SVM is set to 1 for all the experiments.

3.2 Experimental Results

Figs. 1-3 show the results of the GCM dataset and Figs. 4-6 show the results of the ALL dataset. We have the following observations:

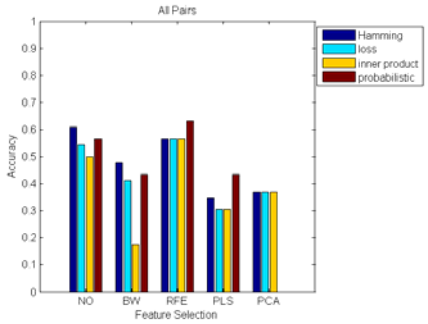

Fig. 1 GCM data, all-pairs coding.

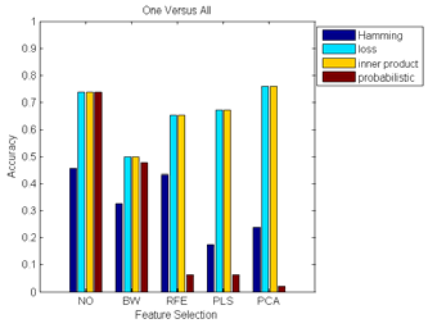

Fig. 2 GCM data, one-versus-all coding.

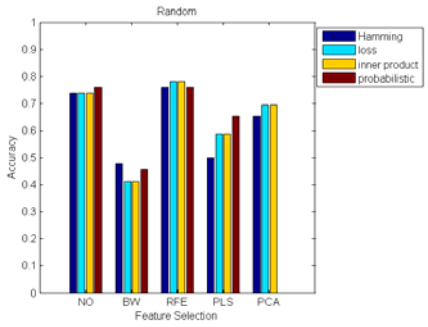

Fig. 3 GCM data, random coding.

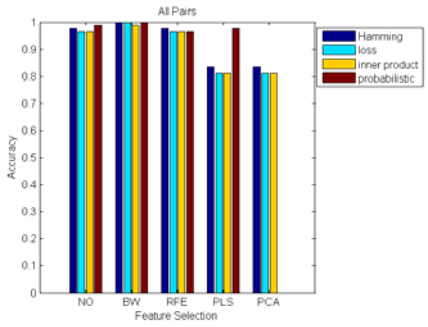

Fig. 4 ALL data, all-pairs coding.

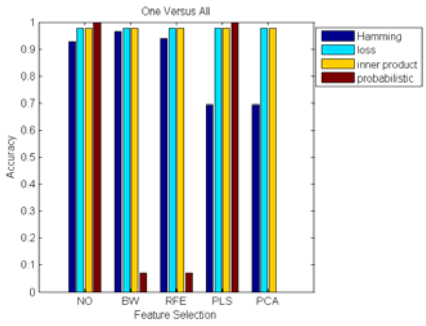

Fig. 5 ALL data, one-versus-all coding.

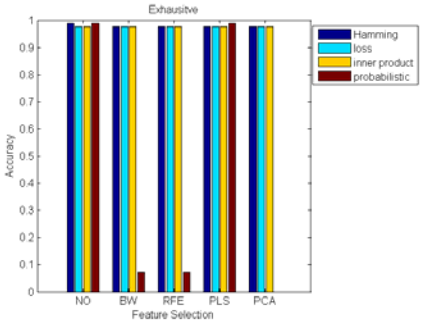

Fig. 6 ALL data, exhaustive coding.

- The ECOC coding strategies generally outperform the other coding strategies. The highest accuracy on the GCM data is achieved by random coding. Combining with loss based and inner product decoding functions and RFE, a 78% testing accuracy has been obtained. On the ALL data, exhaustive coding has achieved almost perfect accuracy for most decoding functions and feature selection methods. There are some exceptions on probabilistic decoding function. This can be attributed to the ability of ECOC to correct errors for weak base classifiers.
- The AP coding strategy works quite well with the ALL data, but it is the worst coding strategy with the GCM data. All testing accuracies on GCM data are below

70%. Learning from Table 1 we know that some classes of the GCM data are very small. It is hard for base classifiers to perform well on these pairs of small cancer classes. A lot of errors may occur for base classifiers, thus the multiclass classification accuracy is degenerated.

- The probabilistic decoding function is very sensitive to coding strategies and feature selections. It fails to work with PCA. It also fails when OVA, exhaustive coding strategies and BW, and RFE feature selections are used on the ALL data. Meanwhile, it achieves 100% accuracy on the ALL data when AP and BW are used, etc. It is known that fitting sigmoid parameters by solving (7) is sensitive to the distribution of samples of two classes. So the unequal distribution of classes of microarray data may lead to the failure of probabilistic decoding function.
- The hamming distance decoding function is not suitable with OVA. It is because many ties will happen when the base classifiers do not give enough high prediction confidence and they are just solved by random assignment. It is better to integrate prediction confidence when OVA is used. However, hamming distance decoding works well with AP and ECOC. This is because the base classifiers of AP usually have high prediction confidence and ECOC has the ability to correct errors if base classifiers are weak. It is noticed that loss based and inner product decoding give very similar results.
- Feature selection by BW ratios performs poorly with GCM data but rather well with ALL data. This is consistent with the results by [2]. BW does not tell information about the class labels so it may select genes that only contain information of several classes without regards to the rest. It is also noticed that results are usually good when no feature selection is used.
- PCA outperforms PLS on the GCM data except using the probabilistic decoding function. However, it has been validated that PLS is usually a better dimension reduction method than PCA [11-12]. It is known from the experiments that the PLS components extracted from GCM data is only 19 while the number of PCA components is 143. We then deduce that the small sizes of some classes in GCM data prevent the PLS to extract enough components so that some information is lost. On ALL data, PLS performs better than PCA.

4 Conclusions

The output coding scheme from machine learning has been successfully applied to multiclass microarray classification in this paper. Usages of different coding matrices, decoding functions and feature selection methods have been discussed. It has been shown that a good coding matrix can lead to high accuracy of multiclass microarray classification. Better coding strategies are required to further improve the performance of the output coding scheme.

Though gene ranking and dimension reduction methods have been shown to be effective for multiclass classification, it is seen that sometimes without feature selection, the results are even better. RFE is good for binary classification but for output coding based multiclass classification, it can only be used to enhance base classifiers. Data over-fitting can easily happen and the variances of outputs would be large especially when class sizes are small. This can degrade the multiclass accuracy in the end. It is better to

use the CV errors of multiclass classification as feedback to select genes. Some algorithms like genetic algorithm could be considered.

References

1. S.B. Cho and H.H. Won. Machine Learning in DNA Microarray Analysis for Cancer Classification. *First Asia-Pacific Bioinformatics Conference*, Adelaide, Australia. Yi-Ping and Phoebe Chen Ed, 2003.
2. T. Li, C. Zhang and M. Ogihara. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20, 15, 2429-2437, 2004.
3. E.L. Allwein, R.E. Schapire, Y. Singer. Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers. *J. Machine Learning Research*, 1, 113-141, 2000.
4. T. Hastie and R. Tibshirani. Classification by pairwise coupling. *The Annuals of Statistics*, 26(2), 451-471, 1998
5. T.G. Dietterich and G. Bakiri. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *J. Artificial Intelligence Research*, 2, 263-286, 1995.
6. F. Ricci and D.W. Aha. Error-Correcting Output Codes for Local Learners. *In Proceedings of the 10th European Conference on Machine Learning*, 1998.
7. J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. *Advances in Large Margin Classifiers*, MIT Press, 1999.
8. I. Guyon, J. Weston, S. Barnhill, V. Vapnik. Gene selection for cancer classification using support vector machines. *Maching learning*, 46, 389-422, 2002.
9. J.A. Wegelin. *A survey of Partial Least Squares (PLS) methods, with emphasis on the two-block case* (Technical Report). Department of Statistics, University of Washin, 2000.
10. G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.
11. D.V. Nguyen and D.M. Rocke. Multi-class cancer classification via partial least squares with gene expression profiles. *Bioinformatics*, 18, 9, 1216-1226, 2002.
12. L. Shen and E.C. Tan. Dimension Reduction Based Penalized Logistic Regression for Cancer Classification Using Microarray Data. *IEEE/ACM Trans. Computational Biology and Bioinformatics*, 2, 166-175, 2005.
13. S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J.P. Mesirov *et al.* Multiclass cancer diagnosis using tumor gene expression signatures. *Proc. Natl Acad. Sci., USA*, 98, 15149–15154, 2001.
14. E.J. Yeoh, M.E. Ross, S.A. Shurtleff, W.K. Williams, D.P. Rami Mahfouz, F.G. Behm, S.C. Raimondi, M.V. Relling, A. Patel, C. Cheng *et al.* Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1, 133-143, 2002.