Joint Classifier and Kernel Design

[1]Balaji Krishnapuram, Lawrence Carin, and Alexander Hartemink

***Abstract*— From a machine learning perspective, the analysis of gene expression data is complicated by the extremely large feature dimensionality. The presence of a large number of irrelevant features—here genes—makes such analysis prone to due to the curse of dimensionality. To overcome this limitation, Support Vector Machines (SVM) are widely employed, since it is well known that they possess good generalization properties even in the presence of irrelevant predictor variables. Motivated by error bounds from computational learning theory, we present a Bayesian generalization of the SVM that jointly learns the optimal classifier and kernel simultaneously from the data. Theoretical and experimental results are provided to show that learning the kernel results in automatic feature selection and hence mitigates the problem of large dimensionality.**

I. Extended Abstract

In the traditional pattern recognition literature, the problem of cancer diagnosis using the gene expression profile of a new tissue sample and a database of previously expression profiles and their diagnoses falls under the general class of *supervised pattern recognition*. Given a database of training samples from N tissues, we have a set of N expression profiles $\boldsymbol{x}^{(i)}$ indexed by $i \in \{1, 2, \ldots, N\}$. Each expression profile $\boldsymbol{x}^{(i)} = [x_1^{(i)}, x_2^{(i)}, \ldots, x_d^{(i)}] \in \mathbb{R}^d$ is a d-dimensional vector representing the measured expression levels of of d genes in the tissue sample. The class membership of each database sample is known and is denoted by $y^{(i)}$. In a two-class case (e.g., the tissues are either cancerous or non-cancerous), we can assume without loss of generality that $y^{(i)} \in \{0, 1\}$. Thus, the training set D consists of N sets of expression profiles and their corresponding class membership labels:

$$D = \left\{ \langle \boldsymbol{x}^{(i)}, y^{(i)} \rangle : \boldsymbol{x}^{(i)} \in \mathbb{R}^d, y^{(i)} \in \{0, 1\} \right\}_{i=1}^{N} \quad (1)$$

Assuming a parametric form for the functional relationship between $\boldsymbol{x}$ and the posterior probability of class membership as $P(y = 1|\boldsymbol{x}) = g_{\boldsymbol{\alpha}}(\boldsymbol{x})$, during the training phase we seek to find the optimal parameters $\boldsymbol{\alpha}$ based on the evidence of the training data, D. In this paper, we consider soft classification functions of the form:

$$P(y = 1|\boldsymbol{x}) = g_{\boldsymbol{\theta},\boldsymbol{\beta}}(\boldsymbol{x}) = \Phi\left(\beta_0 + \sum \beta_i K_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}^{(i)})\right) \quad (2)$$

where $K_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}^{(i)})$ is some symmetric kernel function parameterized by $\boldsymbol{\theta}$, $\boldsymbol{\alpha} = [\boldsymbol{\beta}^T, \boldsymbol{\theta}^T]^T$ are the parameters to be learned during training, and $\Phi(z)$ is the standard Gaussian cumulative distribution function (otherwise known as the probit link function):

$$\Phi(z) = \int_{-\infty}^{z} N(x|0, 1)dx = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} \exp\left(\frac{-x^2}{2}\right) dx \quad (3)$$

[1]Contact Details: Box 90291, EE Dept, Duke University, Durham, NC 27708-0291. balaji@ee.duke.edu

We note that a similar decision function is also used in Support Vector Machines (SVM) which assume that the final classification function is of the form $f_{\boldsymbol{\beta}}(\boldsymbol{x}) = U(\beta_0 + \sum \beta_i K_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}^{(i)}))$, where $U(.)$ is the unit step function. However, in the SVM, $\boldsymbol{\theta}$ is assumed to be fixed, and hence only $\boldsymbol{\beta}$ is learned during training.

The kernel basis function $K_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}^{(i)})$ provides a nonlinear measure of similarity between the gene expression levels of a new unlabeled sample $\boldsymbol{x}$ and a labeled sample from our training database $\boldsymbol{x}^{(i)}$. In this paper, we will use the parameters $\boldsymbol{\theta}$ to represent the scaling factors associated with the genes. Thus, the dimensionality of $\boldsymbol{\theta}$ is d and we can write $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_d]^T \in \mathbb{R}^d$. We have used n-th order polynomial kernels for all the experiments presented here:

$$K_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \left(1 + \sum_{l=1}^{d} \theta_l x_l^{(i)} x_l^{(j)}\right)^n \quad (4)$$

Thus, parameterizing the scaling factors by $\boldsymbol{\theta}$ implies that if $\theta_j = 0$ then our diagnostic classifier does not use any information about the expression level of the j-th gene in the process of making its decision. Theoretical results indicate that we can obtain upper bounds on the error rates of such classifiers that depend on the *margin*, dimensionality and scaling of the feature space, and sparse usage of kernel basis functions. Hence, in order to achieve good generalization performance, we seek to find classifiers (i.e., to find values of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$) that not only predict posterior probabilities of class membership accurately, but also do so with very few non-zero elements in either $\boldsymbol{\theta}$ or $\boldsymbol{\beta}$. Sparsity in $\boldsymbol{\theta}$ implies that the classifier implicitly performs feature selection, while sparsity in $\boldsymbol{\beta}$ implies that it finds a small subset of prototypical samples that are highly representative of the different classes we seek to distinguish.

In a Bayesian framework, learning the optimal $\boldsymbol{\theta}, \boldsymbol{\beta}$ is equivalent to

$$\begin{aligned} \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\beta}} &= \arg\max_{\boldsymbol{\theta},\boldsymbol{\beta}} \log P\left(\boldsymbol{\theta}, \boldsymbol{\beta}|D\right) \\ &= \arg\max_{\boldsymbol{\theta},\boldsymbol{\beta}} \{\log P(\boldsymbol{\theta}) + \log P(\boldsymbol{\beta}) + \log P\left(D|\boldsymbol{\theta}, \boldsymbol{\beta}\right)\} \end{aligned} \quad (5)$$

Assuming that the samples are drawn independently from some true underlying data distribution, it is evident that:

$$P\left(D|\boldsymbol{\theta}, \boldsymbol{\beta}\right) = \prod_{i=1}^{N} \left\{ \begin{array}{l} y^{(i)} g_{\boldsymbol{\theta},\boldsymbol{\beta}}\left(\boldsymbol{x}^{(i)}\right) + \\ \left(1 - y^{(i)}\right)\left(1 - g_{\boldsymbol{\theta},\boldsymbol{\beta}}\left(\boldsymbol{x}^{(i)}\right)\right) \end{array} \right\} \quad (6)$$

As the first step of our Bayesian analysis, we have to specify the prior on the parameters $\boldsymbol{\theta}$, $\boldsymbol{\beta}$ that we want to estimate. We choose to adopt a Laplacian prior on $\boldsymbol{\beta}$, since it is known

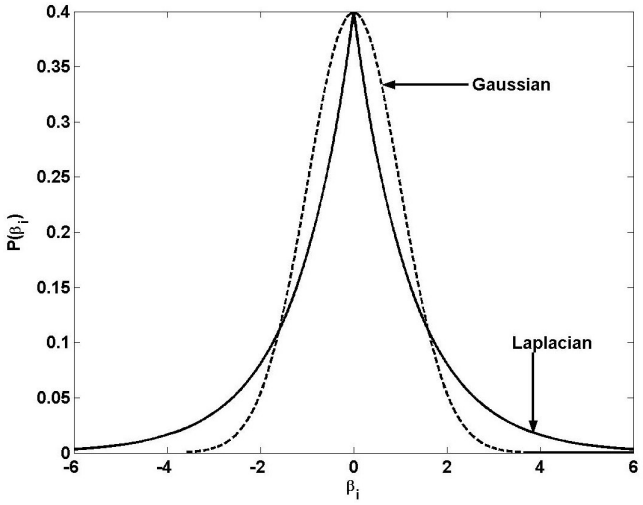

Fig. 1. Sparsity-promoting Laplacian prior: the Laplacian distribution is sharply peaked and not smoothly differentiable about zero, unlike the Gaussian distribution, which is everywhere smoothly differentiable and whose derivative at zero is zero.

TABLE I

ACCURACY OF DIAGNOSTIC CLASSIFICATION: MULTIPLE CLASSIFIERS APPLIED TO TWO BENCHMARK DATASETS (% CORRECT IN LOOCV)

Classifier	**AML/ ALL**	**Colon tumor**
Adaboosting (Decision stumps) [1]	95.8	72.6
SVM (Linear kernel) [1]	94.4	77.4
RVM (No kernel: on feature space) [2]	97.2	88.7
Logistic regression (No kernel: on feature space) [2]	97.2	71.0
Sparse probit regression (Linear kernel)	97.2	91.9
JCFO (Linear kernel)	100.0	96.8

from earlier work that this prior promotes sparseness (making several $\beta_i = 0$) due to its use of the L_1 norm penalty:

$$P(\boldsymbol{\beta}|\eta) = \prod_{i=1}^{M} \frac{\eta}{2} \exp(-\eta\,|\beta_i|) = \left(\frac{\eta}{2}\right)^M \exp(-\eta\,\|\boldsymbol{\beta}\|_1) \quad (7)$$

Figure 1 illustrates this property of a Laplacian prior, and contrasts it with a Gaussian prior, whose derivative at zero is zero. As the figure illustrates, the difference between $P(0)$ and $P(\beta_i)$ for small β_i is much larger for a Laplacian than for a Gaussian. As a result, if we use a Laplacian prior, learning procedures that seek to maximize the posterior would explicitly favor values of β_i that are exactly 0 instead of small values close to 0, thus promoting sparseness in $\boldsymbol{\beta}$.

In estimating the parameter θ_i, we can adopt a prior that is similar to that for β_i but differs in one critical aspect: we must ensure that our algorithms learn $\theta_i \geqslant 0$. The reason for this requirement is somewhat subtle. Essentially, θ_i measures the scaling of the individual features. In the forms of the kernels that we have described in equation (4), using a negative scaling θ_i effectively implies that if we compare two feature vectors using these kernels, similar levels of that particular feature would actually reduce the value of that kernel function between the two observation feature vectors. Though greater similarity of a particular feature's values in two different observations need not necessarily imply that they are *more* similar (in the context of classification, especially when the particular feature is irrelevant for the discrimination between the classes), it can never imply that the observations are somehow *less* similar. Thus, even though θ_i can be exactly zero, it can never be negative. As a result, we consider a prior for θ_i that explicitly makes them non-negative.

$$P(\theta_i|\gamma_2) = \begin{cases} \sqrt{\gamma_2}\exp(-\sqrt{\gamma_2}\theta_i) & \text{if } \theta_i \geqslant 0 \\ 0 & \text{if } \theta_i < 0 \end{cases} \quad (8)$$

Substituting (6), (8), and (7) in (5), the optimization required for our learning task seems to be quite complicated. However, we have recently developed an elegant Expectation Maximization (EM) algorithm which finds the maximum a posteriori (MAP) classifier of (5) efficiently [3].

As shown in Table I, in full leave-one-out cross-validation studies on the two popular benchmark datasets, the accuracy of the diagnostic classification reported by the our Joint Classifier and Feature Optimization (JCFO) algorithm is superior to that reported by other state-of-the-art methods. Similar results were also obtained on several other gene expression datasets that have been analyzed by the authors. Perhaps even more impressively, in every disease that we have analyzed, this method automatically identifies a very small number of genes with $\theta_i \neq 0$ (typically only around 20 out of several thousand genes under consideration) as important for prediction. In several cases, the products of some of the genes that have been identified by this method are in current clinical use as disease markers to diagnose the disease. Most of the identified genes are of known biological relevance to the diagnostic classification being studied. A few novel genes are also highlighted and their function is being further investigated. The proposed method is much sparser in identifying genes as compared to other techniques like the RVM. Further, in cross validation studies the same genes were also consistently identified in almost all the trials. This implies that a greater stability in identifying the features is obtained with the JCFO.

REFERENCES

[1] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini. Tissue classification with gene expression profiles. In *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology (RECOMB)*. Universal Academy Press, Tokyo, Apr. 2000.

[2] B. Krishnapuram, A.J. Hartemink, and L. Carin. Logistic regression and RVM for cancer diagnosis from gene expression signatures. In *Proceedings of the 2002 Workshop on Genomic Signal Processing and Statistics (GENSIPS)*. IEEE Signal Processing Society, Oct. 2002.

[3] B. Krishnapuram, L. Carin, and A.J. Hartemink. Joint classifier and feature optimization for cancer diagnosis using gene expression data. In *Proceedings of the Seventh Annual International Conference on Computational Molecular Biology (RECOMB)*, Apr. 2003.