1 Gene expression analysis: Joint feature selection and classifier design

Balaji Krishnapuram, Lawrence Carin
Department of Electrical Engineering, Duke University
Box 90291, Durham, NC 27708, USA
balaji@ee.duke.edu, lcarin@ee.duke.edu

Alexander Hartemink
Department of Computer Science, Duke University
Box 90129, Durham, NC 27708, USA
amink@cs.duke.edu

Recently developed high-throughput technologies—including oligonucleotide arrays (Lockhart et al., 1996), DNA microarrays (Schena et al., 1995), and SAGE (Velculescu et al., 1995)—enable us to simultaneously quantify the expression levels of thousands of genes in a population of cells. As one application of these technologies, gene expression *profiles* can be generated from a collection of cancerous and non-cancerous tumor tissue samples and then stored in a database. Kernel methods like the *support vector machine* (SVM) and the *relevance vector machine* (RVM) have been shown to accurately predict the disease status of an undiagnosed patient by statistically comparing his or her profile of gene expression levels against a database of profiles from diagnosed patients (Golub et al., 1999; Furey et al., 2000; Alon et al., 1999; Ramaswamy et al., 2001; Li et al., 2002). Despite this early success, the presence of a significant number of irrelevant features—here genes in the profile that are unrelated to the disease status of the tissue—makes such analysis somewhat prone to the *curse of dimensionality.*

Intuitively, overcoming the curse of dimensionality requires that we build classifiers relying on information exclusively from the genes in the profile that are truly relevant to the disease status of the tissue. This problem of identifying the features most relevant to the classification task is known as *feature selection.* In this chapter, we review current methods of feature selection, focusing especially on the many recent results that have been reported in the context of gene expression analysis. Then we present a new Bayesian EM algorithm that jointly accomplishes the classifier design and feature selection tasks. By combining these two problems and solving them together, we identify only those features that are most useful in performing the classification itself. Experimental results are presented on several gene expression datasets. The biological significance of the genes identified by the method is also briefly assessed.

1.1 Common issues in classifying gene expression profiles

For simplicity, we consider in this chapter the problem of diagnosing whether or not a patient has a specific disease, which can be viewed as a binary supervised classification or pattern recognition task. In situations where several diseases are possible, standard strategies for extending binary classifiers to multi-class problems may be used to generalize the analysis presented here.

Notation

Let us assume that we are presented with a database of m profiles each measuring the expression level of N genes, $X = \{\mathbf{x}_i \in \mathbb{R}^N\}_{i=1}^m$. Additionally, we are also provided with the corresponding set of class labels indicating the disease status, $Y = \{y_i \in \{-1, +1\}\}_{i=1}^m$. Assuming a parametric form for the functional relationship between $\mathbf{x}$ and the posterior probability of class membership as $P(y = 1|\mathbf{x}) = f_{\boldsymbol{\beta}}(\mathbf{x})$, during the training phase we seek to find the optimal parameters $\boldsymbol{\beta}$ based on the evidence of the training data, $D = \{X, Y\}$.

Data characteristics

Because of the significant cost and effort required to perform these experiments, currently available databases typically contain fewer than one hundred profiles, though each profile quantifies the expression levels of several thousands of genes. Due to the high dimensionality and the small sample size of the experimental data, it is often possible to find a large number of classifiers that can separate the training data perfectly, but their diagnostic accuracy on unseen test samples is quite poor. In terms of risk, even when we design a classifier to minimize the empirical risk, $\hat{\boldsymbol{\beta}} = \operatorname{argmin}_{\boldsymbol{\beta}} R_{\mathrm{emp}}(f_{\boldsymbol{\beta}})$, the true risk, $R(f_{\hat{\boldsymbol{\beta}}})$, of the resulting classifier remains large. However, as demonstrated later in this chapter, when presented with the expression levels of only a small subset of diagnostically relevant genes, several methods of classifier design achieve equally good generalization. Thus, we may conclude that the choice of feature selection methods is often more important than the choice of classifier for gene expression analysis.

1.2 A review of feature selection methods for kernel machines

Definition

Let us assume that the genes are indexed by the variable $j \in \{1, 2, \ldots, N\}$. We can denote any subset of the genes by S where $S \subset \{1, 2, \ldots, N\}$. We define $X^{(S)}$ to be the database of expression profiles restricted to the genes contained in the subset S. Thus, $X^{(S)} = \{\mathbf{x}_i^{(S)} \in \mathbb{R}^{|S|}\}_{i=1}^m$, where $|S|$ denotes the cardinality of S. In feature selection, we are interested in identifying the subset of genes $\hat{S}$ whose expression levels are most relevant for classification or diagnosis. There are three principal reasons for our interest in feature selection:

Reasons for feature selection

1. We can improve the generalization performance—or out-of-sample accuracy—of our classifier by identifying only the genes that are relevant to the prediction of the disease diagnosis. This effect is attributable to the overcoming of the *curse of dimensionality* alluded to in the previous section.
2. If it is possible to identify a small set of genes that is indeed capable of providing complete discriminatory information, inexpensive diagnostic assays for only a few genes might be developed and be widely deployed in clinical settings.
3. Knowledge of a small set of diagnostically relevant genes may provide important insights into the mechanisms responsible for the disease itself.

Caveats

In considering this last point, it is necessary to clearly distinguish between the relevance of a gene to the biological mechanism underlying the disease and its relevance to building good diagnostic prediction algorithms. On the one hand, not all biologically relevant genes may need to be used when building accurate diagnostic classifiers; on the other hand, high correlation between disease status and gene expression level does not necessarily imply that the expression of that particular gene has somehow *caused* the disease. So diagnostic relevance is neither a necessary nor a sufficient condition for biological relevance. Consequently, genes selected for their diagnostic relevance should be validated for biological relevance by followup studies of the literature or experimental analysis. Nevertheless, the fact remains that good feature selection methods can often serve as excellent guides in identifying small subsets of genes for further investigation.

Three approaches to feature selection

In the current literature three basic approaches to feature selection predominate (Blum and Langley, 1997; Kohavi and John, 1997): *filter*, *wrapper*, and *embedded*. Filter methods consider the problem of feature selection in isolation from the problem of classifier design; typically a subset of features is first selected in a preprocessing step, and then only the selected features are subsequently used to design a classifier. Thus, filter methods are independent of the technique for classifier design and they may be used in conjunction with any such algorithm. In contrast, wrapper methods search through the space of possible feature subsets and measure the quality of a particular subset S by estimating the accuracy of a classifier designed using only the features in S, either directly or indirectly (using theoretical bounds as approximations). Hence, wrapper methods search for optimal feature subsets for use with a specific classifier design algorithm. Finally, in embedded methods, feature selection and classifier design are accomplished jointly. We study several examples of all three methods below.

1.2.1 Filter methods of feature selection

FDR

Perhaps the simplest filtering scheme is to evaluate each feature individually based on its ability to distinguish between the disease categories (*i.e.*, ability to predict

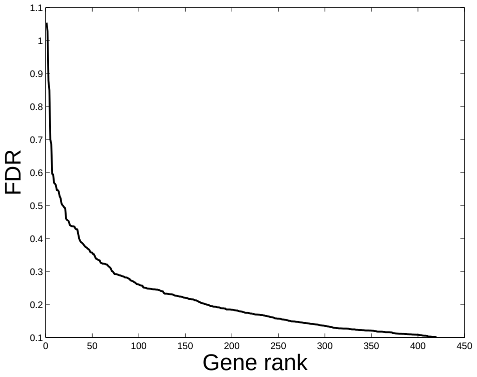

Figure 1.1 Plot of Fisher discriminant ratio of genes with FDR values at least 10% the value of the gene with the highest FDR value. The genes have been sorted by decreasing FDR value.

class labels). We may compute the *Fisher discriminant ratio* (FDR) of each gene j, as:

$$FDR(j) = \frac{(\mu_+^{(j)} - \mu_-^{(j)})^2}{(\sigma_+^{(j)})^2 + (\sigma_-^{(j)})^2} \tag{1.1}$$

where $\mu_+^{(j)}$, $\mu_-^{(j)}$, $\sigma_+^{(j)}$, and $\sigma_-^{(j)}$ represent the class-conditional means and standard deviations, respectively, for the expression level of gene j. Under the assumption that the class-conditional density functions are Gaussian, larger values of FDR(j) suggest that gene j is better able to distinguish between classes when used as the single independent predictor for disease status. Consequently, selecting genes with the highest FDR values is often employed as a simple technique for feature selection.

To study the limitations of this method, let us consider the gene expression data provided by Alon et al. (1999) where the aim is to distinguish colon tumors from normal cells. Figure 1.1 shows the FDR values of genes from this dataset, sorted by decreasing FDR value. Note that the expression levels of over 400 genes are well correlated with the diagnostic class labels, as determined by FDR. This presents far too large a set to reasonably consider if the importance of each of these genes to the disease mechanism must be experimentally validated, or if interactions between the genes are to be explored using automated computational network inference methods.

Redundant features

In looking more closely at these 400 genes, we uncover a great deal of redundancy. For example, consider Figure 1.2, which shows the expression profiles of the third and eleventh ranked genes. These two genes are highly correlated and thus their expression profiles provide similar information. In fact, large subsets of the genes with high FDR values exist that encode essentially the same information from the point of view of classifier design, implying a heavy redundancy of information among the most prominent features. As a result, considering all the genes with high FDR

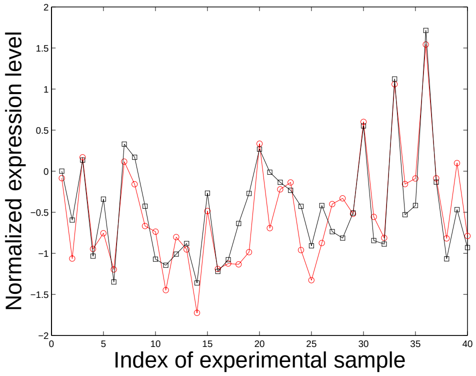

Figure 1.2 Normalized expression levels of genes ranked 3 and 11 according to FDR value. For perceptual clarity, only the expression level on the tumor samples is shown. Again for perceptual clarity, only two genes are depicted here, but the same effect is obtained for large clusters of genes.

values provides little more information than that obtained using the single highest gene. Certainly, from the perspective of understanding the disease mechanism, it is important to distinguish between genes whose expression level is not correlated with the diagnosis (*i.e.*, irrelevant features) and sets of genes which are highly correlated to each other (*i.e.*, redundant features). However, from the perspective of classifier design, both of these provide potential problems. Hence, we may conclude that while FDR may identify a large number of relevant genes, the identified set likely has heavy redundancy. This suggests that more insightful feature selection methods may be able to identify much smaller sets of relevant genes that are nevertheless equally effective in diagnosing the disease status of a patient.

PCA and other projection techniques

One commonly suggested mechanism for addressing this redundancy problem is to first identify a low dimensional space such that the projections of the samples onto that space are linear combinations of the original features. This is accomplished using principal components analysis (PCA) or linear discriminant analysis, and has been used with some success in gene expression analysis, for example in Khan et al. (2001). Unfortunately, since the new predictors are now linear combinations of all of the genes at once, we no longer have the advantage of requiring few experimental measurements for inexpensive clinical deployment and we lose most of the mechanistic insights as well. Similar problems are also associated with several nonlinear extensions of these projection methods.

1.2.2 Wrapper methods of feature selection

The main limitation of methods that select features independently of one another—like FDR—is that they are typically designed under the assumption of independent and thus non-redundant features. This limitation can be largely overcome by considering feature selection as a search over the space of all possible feature subsets. Exploiting the natural partial ordering properties of the space of subsets, we can either start with an empty set and successively add features, or start with the set of all features and successively remove them. The former approach is referred to as forward selection while the latter is referred to as backward elimination; note that a combination of the two approaches is also possible.

Forward selection

As an example of forward feature selection, using any classifier design algorithm, we might first look for the single most discriminative feature. We could then look for the single additional feature that gives the best class discrimination *when considered along with the first feature* (note the difference between this approach and that of FDR). We could keep augmenting the feature set iteratively in this greedy fashion until cross-validation error estimates are minimized; if adding additional features leads to lower estimates of generalization performance, we stop. As an example, Xiong et al. (2001) use this kind of greedy forward feature selection approach along with a Fisher linear discriminant classifier to analyze gene expression data for tumor classification. Note that this approach requires a large number of classifiers to be built, but the cost of building each such classifier is quite small since the number of features considered is not very large at any point.

Backward elimination

In backward elimination, we start by building a classifier using all the features. Assessing the relative importance of the features in the resulting classifier, we iteratively eliminate the least important features and again build a classifier based on the remaining feature subset. Since the final subset may be only a small fraction of the size of the original feature set, we can accelerate the elimination process by removing large sets of irrelevant features in each iteration. This reduces the number of intermediate classifiers that need to be built during the process. Even with such an optimization, however, each of these iterations may still require a large amount of effort since the feature set remains large for several iterations. Therefore kernel classifiers are a natural choice in this framework since their computational requirements scale well to high feature dimensionality.

By varying the algorithm used for classifier design and the criterion used to measure feature relevance, a family of related backward elimination methods have been developed for gene expression analysis; we use the remainder of this section to examine a number of them.

Wrapper methods based on minimization of error bounds

The leave-one-out error L is an unbiased estimator of the generalization performance of classifiers. Hyperplane classifiers like the SVM are often of the form:

$$f(\mathbf{x}; \mathbf{w}, b) = (\mathbf{w} \cdot \mathbf{x}) + b \tag{1.2}$$

For classifiers of this form, the following radius/margin bound for L is well known (Vapnik, 1995):

$$L \leq 4R^2 \|\mathbf{w}\|_2^2 \tag{1.3}$$

where R is the radius of the smallest sphere in the kernel induced feature space that contains all the data, and $\mathbf{w}$ specifies the hyperplane classifier identified by the SVM in that space. Assuming that R changes little with the choice of features, minimizing the margin $\|\mathbf{w}\|_2$ should result in a tighter bound for L. Guyon et al. (2002) use this intuition as the basis for assessing the importance of the features at each iteration of their wrapper method. The result is an algorithm they term *recursive feature elimination* (RFE) that performs feature selection by iteratively training an SVM classifier with the current set of genes and removing the genes with the smallest weight in the resulting hyperplane. In related work, Weston et al. (2000) attempt feature selection by minimizing (1.3) using a gradient descent algorithm on the feature scaling factors rather than eliminating them. Retaining the backward elimination approach of Guyon et al. (2002), Rakotomamonjy (2003) discusses two generalizations of these algorithms. First, use of a tighter span bound on L as compared to (1.3) is considered. Second, for both (1.3) and the span bound, the magnitude of the gradient of the error bounds w.r.t. the feature scale is used as a measure of feature relevance. Intuitively, removing the features with the smallest gradient has the least effect on the generalization error bound. Hence in each iteration an SVM classifier is computed, the features are ranked according to the gradients of the error bounds and the least significant features are removed.

Comparison of wrapper methods

Among these bound minimization algorithms for feature selection with the SVM, the RFE has empirically been observed to achieve the best results on classification tasks using gene expression data (Rakotomamonjy, 2003). Zhu and Hastie (2003) have shown that recursive feature elimination with a *penalized kernel logistic regression* (PKLR) classifier in place of an SVM classifier achieves classification accuracy equal to that of RFE with the SVM, but with two additional benefits: RFE with the PKLR tends to find an even more parsimonious feature subset than RFE with the SVM, and this approach also provides posterior probabilities of class membership. In other related work, Ambroise and McLachlan (2002) have found that the RFE achieves a maximum of about 3% improvement in classification error rates as compared to equivalent forward selection methods using the SVM classifier.

1.2.3 Embedded methods of feature selection

Zero-norm minimization

For constructing hyperplane classifiers of the form (1.2), Weston et al. (2003) provide an algorithm to approximately solve the following problem:

$$\begin{aligned} \hat{\mathbf{w}} &= \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^N} \lambda \|\mathbf{w}\|_0 + \|\boldsymbol{\xi}\|_0 \\ \text{SUBJECT TO:} \quad & y_i\left((\mathbf{w}\cdot\mathbf{x}_i)+b\right) \geq 1-\xi_i \end{aligned} \tag{1.4}$$

Recall that the l_0 norm of a vector—denoted as $\|\cdot\|_0$ above—is equal to the number of non-zero elements in it. Thus, the algorithm of Weston et al. (2003) tries to jointly minimize a weighted sum of the number of features used in the hyperplane classifier and the number of misclassifications on the training data. As a consequence, feature selection is incorporated fundamentally into the algorithm for classifier design. Though the choice of the l_0 norm constitutes an extreme example of feature selection enforced as part of classifier design, several recently developed algorithms for classifier design have incorporated regularizers based instead on the l_1 norm: $\|\mathbf{w}\|_1 = \sum |\mathrm{w}_i|$, which is also often referred to as the lasso penalty.

Theoretical properties of l_1 regularizers

The family of l_1 regularized algorithms share interesting theoretical properties and relationships. A theorem from Mangasarian (1999) establishes the equivalence between l_p margin maximization and l_q distance maximization where $\frac{1}{p} + \frac{1}{q} = 1$. Using this result, Rosset et al. (2003) have shown that one-norm SVM, exponential boosting, and l_1 regularized logistic regression all converge to the *same* non-regularized classifier in the limit as $\lambda \to 0$. This classifier is shown to be a hyperplane that maximizes the l_∞ distance from the closest points on either side. Friedman et al. (2004) consider situations that are not uncommon in gene expression analysis: *e.g.*, $m = 100$ and $N = 10,000$. They argue that in a sparse scenario where only a small number of true coefficients w_i are non-zero, the l_1 margin regularizer works better than the normal l_2 margin used in penalized logistic regression, SVM, *etc.*; in the non-sparse scenario, neither regularizer fits coefficients well due to the curse of dimensionality. Based on these observations, they propose the *bet on sparsity principle* for high dimensional problems which encourages using the l_1 penalty.

One-norm SVM

One example of an l_1 regularized method is the one-norm SVM of Fung and Mangasarian (2002):

$$\begin{aligned} \hat{\mathbf{w}} &= \operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^N} \lambda \|\mathbf{w}\|_1 + \|\boldsymbol{\xi}\|_1 \\ \text{SUBJECT TO:} &\quad y_i \left((\mathbf{w} \cdot \mathbf{x}_i) + b \right) \geq 1 - \xi_i \end{aligned} \tag{1.5}$$

Sparse logistic regression

Replacing the hinge loss function of the SVM in (1.5) with the logistic loss function leads to an objective function that is the *maximum a posteriori* (MAP) solution of logistic regression with a Laplacian prior (Krishnapuram et al., 2003b; Roth, 2003):

$$\begin{aligned} \hat{\mathbf{w}} &= \operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^N} \lambda \|\mathbf{w}\|_1 + \sum_{i=1}^{m} \log\left(1 + \exp\left(-y_i f\left(\mathbf{x}_i; \mathbf{w}\right)\right)\right) \\ &= \operatorname*{argmax}_{\mathbf{w} \in \mathbb{R}^N} \frac{1}{\lambda} \exp\left(-\lambda \|\mathbf{w}\|_1\right) \prod_{i=1}^{m} \frac{1}{\left(1 + \exp\left(-y_i f\left(\mathbf{x}_i; \mathbf{w}\right)\right)\right)} \end{aligned} \tag{1.6}$$

Sparse probit regression, RVM, JCFO

Other closely related algorithms for sparse hyperplane classifier design may be obtained by changing the logistic regression to the probit regression (Figueiredo, 2003), or by changing the Laplacian prior to a Student-t prior (Tipping, 2001). The latter is the basis of the *relevance vector machine* (RVM). All these variations provide comparable feature selection and classification accuracy. While all the embedded methods described in this section can be used to perform joint feature selection and classifier design in the context of simple hyperplane classifiers, they are not able to accomplish joint feature selection and classifier design in the context of nonlinear kernel classifiers. In the next section we derive a new algorithm for *joint classifier and feature optimization* (JCFO) that extends these methods to accomplish joint feature selection and classifier design in the context of nonlinear kernel classifiers.

1.3 The joint classifier and feature optimization algorithm

Basic intuition behind JCFO

The basic intuition behind our approach to solving the feature selection and classifier design problems jointly is to extend the set of parameters to be learned: we will estimate not only the weight parameters $\boldsymbol{\alpha}$ associated with basis functions but also a vector of (non-negative) scaling factors $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_N]^T$ associated with the N features. Consequently, we consider functions of the form

$$f(\mathbf{x}) = \Phi\left(\boldsymbol{\alpha}^T \boldsymbol{h}_{\boldsymbol{\theta}}(\mathbf{x})\right) = \Phi\left(\alpha_0 + \sum \alpha_i \; \psi_i(\mathbf{x}, \boldsymbol{\theta})\right) \tag{1.7}$$

where: $\Phi(z)$ is the probit link function; $\boldsymbol{\alpha}$ is a vector of weights $[\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_k]^T$; $\boldsymbol{h}_{\boldsymbol{\theta}}(\mathbf{x}) = [1, \psi_1(\mathbf{x}, \boldsymbol{\theta}), \ldots, \psi_k(\mathbf{x}, \boldsymbol{\theta})]^T$; and $\psi_i(\mathbf{x}, \boldsymbol{\theta}) : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$ are (possibly nonlinear) basis functions. Please note that $\Phi(z)$ is *not* used in this chapter as the map into feature space induced by some kernel; rather $\Phi(z) = (2\pi)^{-1/2} \int_{-\infty}^{z} \exp\left(-x^2/2\right) dx$.

Kernel basis functions

Although our formulation allows arbitrary basis functions, we shall focus on the important case where $\psi_i(\mathbf{x}, \boldsymbol{\theta}) = k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}_i)$ is some symmetric Mercer kernel function (Cristianini and Shawe-Taylor, 2000), parameterized by $\boldsymbol{\theta}$. Accordingly, the dimension of both $\boldsymbol{\alpha}$ and $\boldsymbol{h}_{\boldsymbol{\theta}}(\mathbf{x})$ is $m + 1$. The only condition we place on the kernel $k_{\boldsymbol{\theta}}$ is that its dependence on $\boldsymbol{\theta}$ is such that smaller values of θ_j correspond to smaller influence of $x^{(j)}$ and $x_i^{(j)}$ in $k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}_i)$; in particular, if $\theta_j = 0$, then $k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}_i)$ must not depend on $x^{(j)}$ or $x_i^{(j)}$. Examples of such functions are scaled Gaussian kernels, $k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}_i) = \exp\{-(\mathbf{x} - \mathbf{x}_i)^T \operatorname{diag}(\boldsymbol{\theta}) (\mathbf{x} - \mathbf{x}_i)\}$, and n-th order polynomial kernels, $k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}_i) = (1 + \mathbf{x}^T \operatorname{diag}(\boldsymbol{\theta}) \, \mathbf{x}_i)^n$.

Bayesian learning of parameters

Our algorithm may be characterized as a Bayesian approach to learning the weight parameters $\boldsymbol{\alpha}$ and the scaling $\boldsymbol{\theta}$. Accordingly, in the next section we define priors over both $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ that reflect our *a priori* belief that most of the elements of these vectors are identically zero. Subsequently, we jointly estimate the maximum *a posteriori* (MAP) values of $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ using an expectation maximization algorithm.

Related work

It should be noted that Seeger (2000) and Williams and Barber (1998) also attempt similar aims of joint feature scale identification and classifier design in a Bayesian setting; however, those methods have not been applied to problems in gene expression analysis. Despite similar objectives, the three methods differ significantly in their algorithmic details; we only derive the JCFO hereafter. Though we do not have experimental evidence to prove it, we expect all three methods to perform comparably and enjoy similar benefits.

1.3.1 Sparsity-promoting priors and their hierarchical decomposition

Laplacian priors promote sparsity

We seek to find classifiers (*i.e.*, to estimate $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$) that not only predict class probabilities accurately, but also do so with few non-zero elements in either $\boldsymbol{\alpha}$ or $\boldsymbol{\theta}$. When using a kernel basis function formulation, sparsity in $\boldsymbol{\alpha}$ corresponds to finding a small subset of training samples that are representative of the classes (as in Tipping (2001); Figueiredo and Jain (2001)), while sparsity in $\boldsymbol{\theta}$ corresponds to implicit feature selection. As already pointed out in 1.2.3, it is well known that sparsity can be encouraged by a Laplacian prior (see for instance Tibshirani (1996)).

For the prior on $\boldsymbol{\alpha}$, we have $p(\boldsymbol{\alpha}|\eta) \propto \exp(-\eta \|\boldsymbol{\alpha}\|_1) = \prod_j \exp(-\eta |\alpha_j|)$, where η is a hyperparameter whose choice will be addressed below. In the case of $\boldsymbol{\theta}$, since the parameters are all non-negative, we have $p(\boldsymbol{\theta}|\nu) \propto \exp(-\nu \|\boldsymbol{\theta}\|_1) = \prod_l \exp(-\nu\theta_l)$, for all $\theta_l \geq 0$, zero otherwise.

Hierarchical priors

Maximum *a posteriori* (MAP) estimates of $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ cannot be found in closed form. To circumvent this difficulty, we consider the following two-level hierarchical model. Each α_i is given a zero-mean Gaussian prior with its own variance τ_i: $p(\alpha_i|\tau_i) = N(\alpha_i|0,\tau_i)$. Further, the variances τ_i have independent exponential hyperpriors, $p(\tau_i|\gamma_1) \propto \exp\left(-\gamma_1\tau_i/2\right)$, for $\tau_i \geq 0$. The effective prior can be obtained by integrating out τ_i:

$$p(\alpha_i|\gamma_1) = \int_0^\infty p(\alpha_i|\tau_i)p(\tau_i|\gamma_1)d\tau_i \propto \exp(-\sqrt{\gamma_1}\,|\alpha_i|) \tag{1.8}$$

showing that a Laplacian prior is equivalent to a two-level hierarchical model characterized by zero-mean Gaussian priors with independent exponentially distributed variances. For each parameter θ_i, since it is non-negative, we adopt non-negative Gaussian priors

$$p(\theta_i|\rho_i) = \begin{cases} 2N(\theta_i|0,\rho_i) & \text{if } \theta_i \geqslant 0 \\ 0 & \text{if } \theta_i < 0 \end{cases} \tag{1.9}$$

and again the ρ_i have independent exponential hyperpriors: $p(\rho_i|\gamma_2) \propto \exp(-\gamma_2\rho_i/2)$, for $\rho_i \geqslant 0$. The effective prior on θ_i is thus exponential, as desired:

$$p(\theta_i|\gamma_2) \propto \begin{cases} \exp(-\sqrt{\gamma_2}\,\theta_i) & \text{if } \theta_i \geqslant 0 \\ 0 & \text{if } \theta_i < 0 \end{cases} \tag{1.10}$$

1.3.2 The JCFO algorithm

Missing data interpretation of probit link

The hierarchical decomposition of the Laplacian priors just described opens the door to the use of an EM algorithm for computing the estimates of $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ by treating the τ_i and ρ_i as missing data. Furthermore, the probit link allows an interpretation in terms of hidden variables that facilitates the derivation of such an EM algorithm (Figueiredo and Jain, 2001; Albert and Chib, 1993). Specifically, let $z(\mathbf{x},\boldsymbol{\alpha},\boldsymbol{\theta}) = \boldsymbol{\alpha}^T \boldsymbol{h}_{\boldsymbol{\theta}}(\mathbf{x}) + \epsilon$, where ϵ is a zero-mean unit-variance Gaussian random variable. If the classifier is defined as $y = \text{sign}(z(\mathbf{x},\boldsymbol{\alpha},\boldsymbol{\theta}))$, then we recover the probit model, since

$$P(y=1|\mathbf{x}) = P\left(\boldsymbol{\alpha}^T \boldsymbol{h}_{\boldsymbol{\theta}}(\mathbf{x}) + \epsilon > 0\right) = \Phi\left(\boldsymbol{\alpha}^T \boldsymbol{h}_{\boldsymbol{\theta}}(\mathbf{x})\right) \tag{1.11}$$

Given the data $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$, consider the corresponding vector of missing variables $\mathbf{z} = [z_1, \ldots, z_m]^T$, as well as the vectors of missing variables $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_{m+1}]^T$ and $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_N]^T$. If $\mathbf{z}$, $\boldsymbol{\tau}$, and $\boldsymbol{\rho}$ were known, we would have an easier estimation problem for $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$: We would effectively have the observation model $\mathbf{z} = \boldsymbol{H}_{\boldsymbol{\theta}}\boldsymbol{\alpha} + \boldsymbol{\epsilon}$, with Gaussian priors on $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ (variances given by $\boldsymbol{\tau}$ and $\boldsymbol{\rho}$), where $\boldsymbol{H}_{\boldsymbol{\theta}} = [\boldsymbol{h}_{\boldsymbol{\theta}}(\mathbf{x}_1), \boldsymbol{h}_{\boldsymbol{\theta}}(\mathbf{x}_2), \ldots, \boldsymbol{h}_{\boldsymbol{\theta}}(\mathbf{x}_m)]^T$ is the design matrix, and $\boldsymbol{\epsilon}$ is a vector of i.i.d. zero-mean unit-variance Gaussian random variables. This suggests using an EM algorithm to find a local maximum of the posterior $p(\boldsymbol{\alpha}, \boldsymbol{\theta}|D)$.

EM algorithm

The EM algorithm will produce a sequence of estimates $\widehat{\boldsymbol{\alpha}}^{(t)}$ and $\widehat{\boldsymbol{\theta}}^{(t)}$ by alternating between two steps:

- **E-step:** Conditioned on D and on the current estimates $\widehat{\boldsymbol{\alpha}}^{(t)}, \widehat{\boldsymbol{\theta}}^{(t)}$, compute the expected value of the complete log-posterior, $p(\boldsymbol{\alpha}, \boldsymbol{\theta}|D, \mathbf{z}, \boldsymbol{\tau}, \boldsymbol{\rho})$, denoted as $Q\left(\boldsymbol{\alpha}, \boldsymbol{\theta} \;\middle|\; \widehat{\boldsymbol{\alpha}}^{(t)}, \widehat{\boldsymbol{\theta}}^{(t)}\right)$:

$$Q\left(\boldsymbol{\alpha}, \boldsymbol{\theta} \;\middle|\; \widehat{\boldsymbol{\alpha}}^{(t)}, \widehat{\boldsymbol{\theta}}^{(t)}\right) = \int p\left(\mathbf{z}, \boldsymbol{\tau}, \boldsymbol{\rho} \;\middle|\; D, \widehat{\boldsymbol{\alpha}}^{(t)}, \widehat{\boldsymbol{\theta}}^{(t)}\right) \log p(\boldsymbol{\alpha}, \boldsymbol{\theta}|D, \mathbf{z}, \boldsymbol{\tau}, \boldsymbol{\rho})\, d\mathbf{z}\, d\boldsymbol{\tau}\, d\boldsymbol{\rho}$$

- **M-step:** Update the estimates to $\left(\widehat{\boldsymbol{\alpha}}^{(t+1)}, \widehat{\boldsymbol{\theta}}^{(t+1)}\right) = \underset{\boldsymbol{\alpha}, \boldsymbol{\theta}}{\operatorname{argmax}}\; Q\left(\boldsymbol{\alpha}, \boldsymbol{\theta} \;\middle|\; \widehat{\boldsymbol{\alpha}}^{(t)}, \widehat{\boldsymbol{\theta}}^{(t)}\right)$.

E-step

As shown in Appendix 1.7, the E-step reduces to the following three analytical expressions:

$$v_i \equiv \mathbf{E}\left[z_i \;\middle|\; D, \widehat{\boldsymbol{\alpha}}^{(t)}, \widehat{\boldsymbol{\theta}}^{(t)}\right] = \boldsymbol{h}_{\boldsymbol{\theta}}^T(\mathbf{x}_i)\widehat{\boldsymbol{\alpha}}^{(t)} + \frac{y_i\; N\left(\boldsymbol{h}_{\boldsymbol{\theta}}^T(\mathbf{x}_i)\widehat{\boldsymbol{\alpha}}^{(t)} \,\middle|\, 0, 1\right)}{\frac{(1+y_i)}{2} - y_i\; \Phi\left(-\boldsymbol{h}_{\boldsymbol{\theta}}^T(\mathbf{x}_i)\widehat{\boldsymbol{\alpha}}^{(t)}\right)} \tag{1.12}$$

$$\omega_i \equiv \mathbf{E}\left[\tau_i^{-1} \;\middle|\; D, \widehat{\boldsymbol{\alpha}}_i^{(t)}, \gamma_1\right] = \gamma_1 \left|\widehat{\boldsymbol{\alpha}}_i^{(t)}\right|^{-1} \tag{1.13}$$

$$\delta_i \equiv \mathbf{E}\left[\rho_i^{-1} \;\middle|\; D, \widehat{\boldsymbol{\theta}}_i^{(t)}, \gamma_2\right] = \gamma_2 \left(\widehat{\boldsymbol{\theta}}_i^{(t)}\right)^{-1} \tag{1.14}$$

Defining vector $\boldsymbol{v} = [v_1, \ldots, v_m]^T$ and matrices $\boldsymbol{\Omega} = \operatorname{diag}(\omega_1, \ldots, \omega_{m+1})$, and $\boldsymbol{\Delta} = \operatorname{diag}(\delta_1, \ldots, \delta_N)$, the Q function can be written as

$$Q\left(\boldsymbol{\alpha}, \boldsymbol{\theta} \;\middle|\; \widehat{\boldsymbol{\alpha}}^{(t)}, \widehat{\boldsymbol{\theta}}^{(t)}\right) = -\boldsymbol{\alpha}^T \boldsymbol{H}_{\boldsymbol{\theta}}^T \boldsymbol{H}_{\boldsymbol{\theta}} \boldsymbol{\alpha} + 2\,\boldsymbol{\alpha}^T \boldsymbol{H}_{\boldsymbol{\theta}}^T \boldsymbol{v} - \boldsymbol{\alpha}^T \boldsymbol{\Omega}\, \boldsymbol{\alpha} - \boldsymbol{\theta}^T \boldsymbol{\Delta} \boldsymbol{\theta} \tag{1.15}$$

M-step

As for the M-step, since $\boldsymbol{H}_{\boldsymbol{\theta}}$ is generally neither linear nor quadratic in $\boldsymbol{\theta}$, Q cannot be maximized analytically w.r.t. $\boldsymbol{\theta}$. Moreover, the optimizations w.r.t. $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ cannot be pursued independently. However, we observe that given any $\boldsymbol{\theta}$, the optimal $\boldsymbol{\alpha}$ is simply

$$\widehat{\boldsymbol{\alpha}}_{\boldsymbol{\theta}}^{(t+1)} = (\boldsymbol{\Omega} + \boldsymbol{H}_{\boldsymbol{\theta}}^T \boldsymbol{H}_{\boldsymbol{\theta}})^{-1} \boldsymbol{H}_{\boldsymbol{\theta}}^T \boldsymbol{v} = \boldsymbol{\kappa}(\boldsymbol{I} + \boldsymbol{\kappa} \boldsymbol{H}_{\boldsymbol{\theta}}^T \boldsymbol{H}_{\boldsymbol{\theta}} \boldsymbol{\kappa})^{-1} \boldsymbol{\kappa} \boldsymbol{H}_{\boldsymbol{\theta}}^T \boldsymbol{v} \tag{1.16}$$

where $\boldsymbol{\kappa} = \gamma_1^{-1/2} \operatorname{diag}\left(\sqrt{\left|\widehat{\alpha}_1^{(t)}\right|}, \ldots, \sqrt{\left|\widehat{\alpha}_{m+1}^{(t)}\right|}\right)$ is a matrix introduced to enable a stable numerical implementation, since the sparsity-promoting properties of the hierarchical priors will drive several of the α_i to zero. Since we can maximize analytically w.r.t. $\boldsymbol{\alpha}$, we are left with the maximization w.r.t. $\boldsymbol{\theta}$ of $Q(\widehat{\boldsymbol{\alpha}}_{\boldsymbol{\theta}}^{(t+1)}, \boldsymbol{\theta}|\widehat{\boldsymbol{\alpha}}^{(t)}, \widehat{\boldsymbol{\theta}}^{(t)})$.

Here, we are forced to employ numerical optimization to obtain $\widehat{\boldsymbol{\theta}}^{(t+1)}$; in the results presented below, we use conjugate gradient. Note that our model assumes that each $\theta_i \geq 0$; this can be accomplished by the reparameterization $\theta_i = \exp(\zeta_i)$, and solving for each ζ_i instead.

Computational complexity

The computational complexity of the algorithm just outlined remains moderate for problems with m on the order of a few hundred and N on the order of a few thousand, due to the use of some simple approximations (Krishnapuram et al., 2003a). For all of the gene expression datasets analyzed in the next section, we could complete the training of a classifier in under half an hour on a 800 MHz Pentium III machine with unoptimized MATLAB code. The computational bottleneck is clearly the matrix inversion in (1.16), which becomes impractical for large m; this problem is endemic among kernel methods.

1.4 Experimental studies comparing the methods

Two types of experiments

In order to assess the success of the JCFO and to compare its performance against other popular classifiers we performed two types of experiments. In the first experiment, we synthetically generated data with a large number of irrelevant features, in order to study the effect of overfitting on several algorithms. In the second series of experiments, we used measured gene expression data and compared the methods based on their cross-validation estimate of error rates. The genes identified by the JCFO were also evaluated against the known literature for their biological significance. These experiments are outlined below in more detail.

In keeping with standard practice on kernel classifiers, in all experiments the datasets were mean centered and normalized to unit variance. The hyperparameters γ_1 and γ_2 were adjusted by using a hold-out test set of 10% of the data. The chosen values were then used with the entire dataset to obtain the classifiers.

1.4.1 Effect of irrelevant predictor variables on kernel classifiers

To assess the extent to which different state-of-the-art kernel classifiers are affected by the presence of irrelevant predictor variables, we generated N-dimensional Gaussian data from two classes with means $\mu_1 = -\mu_2 = [1/\sqrt{2}, 1/\sqrt{2}, 0, 0, \ldots, 0]^T$ with both covariances identity matrices. Independent of N, the Bayes error rate is $\Phi(-1|0,1) \simeq 0.1587$ and the optimal classifier uses only the first two variables. All algorithms were trained with 100 samples per class, and the accuracy of the learned classifier was tested on an independent test set of 1000 samples. Results were averaged over 20 random draws of the training set, and the procedure repeated for several feature dimensions N. The average error rates plotted in Figure 1.3 show that the JCFO algorithm is far more resistant to the presence of irrelevant variables than the other methods considered: SVM, RVM, and sparse probit (Figueiredo and Jain, 2001). Among the kernel methods compared in this experiment, only the JCFO has the ability to identify the scale of the input features, so it is quite understandable that the JCFO is largely immune to the presence of irrelevant features. Furthermore, for almost all random draws of the training set, the JCFO reported non-zero scaling factors θ_i for only the two relevant features. The performance degradation of the other methods in the presence of irrelevant features shows explicitly that feature selection is crucial, even for large margin techniques like the SVM.

1.4.2 Cancer diagnosis using gene expression data

Benchmarks

Next, we considered two publicly available gene expression datasets that have been analyzed by several authors, though each has adopted a slightly different experimental setup. These datasets are characterized by small training sets (a few tens) but very high feature dimensionality (several thousands). The first one, originally studied by Golub et al. (1999), contains expression values of $N = 7129$ genes from $m = 72$ samples of two classes of leukemia: 47 of acute myeloid leukemia (AML) and 25 of acute lymphoblastic leukemia (ALL). The second dataset, originally analyzed by Alon et al. (1999), contains expression levels of $N = 2000$ genes from 40 tumor and 22 normal colon tissues. Both datasets contain a large number of redundant and irrelevant features (genes). Strong feature selection is thus both possible and desirable.

Diagnostic accuracies of diagnostic classification schemes are commonly assessed using a cross-validation scheme. In an leave-one-out cross-validation (LOOCV), the accuracy of diagnostic prediction on each sample is assessed based on classifiers built using the remaining $m-1$ samples as a training set. Table 1.1 reports the LOOCV results for the JCFO and several other learning methods, including Adaboosting, the SVM, the RVM, logistic regression, and sparse probit regression.

Results from the literature

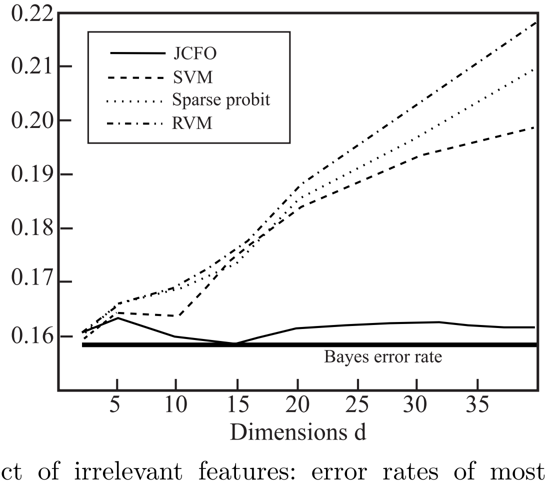

Figure 1.3 Effect of irrelevant features: error rates of most kernel classifiers increase as irrelevant features are introduced but the JCFO is largely immune.

Table 1.1 LOOCV accuracy of cancer diagnosis based on gene expression data for several classifiers (percent correct; a higher number is better).

Classifier	**AML/ALL**	**Colon tumor**
Adaboost (decision stumps) (Ben-Dor et al., 2000)	95.8	72.6
SVM (quadratic kernel) (Ben-Dor et al., 2000)	95.8	74.2
SVM (linear kernel) (Ben-Dor et al., 2000)	94.4	77.4
RVM (linear kernel)	94.4	80.6
RVM (no kernel: on feature space)	97.2	88.7
Logistic regression (no kernel: on feature space)	97.2	71.0
Sparse probit regression (quadratic kernel)	95.8	84.6
Sparse probit regression (linear kernel)	97.2	91.9
Sparse probit regression (no kernel: on feature space)	97.2	85.5
JCFO (quadratic kernel)	98.6	88.7
JCFO (linear kernel)	100.0	96.8

Table 1.2 Cross-validation estimates of accuracy of cancer diagnosis on colon tumor gene expression data for several classifiers, reproduced from results reported in the literature (percent corect; a higher number is better; see text for details of specific experimental setup in each case).

Classifier	**Colon tumor**
RVM (no kernel: on feature space) (Li et al., 2002)	85.4
SVM (RFE) (Weston et al., 2003)	87.5
Zero-Norm minimization (Weston et al., 2003)	88.9
SVM (Radius/margin gradient) (Rakotomamonjy, 2003)	88.9

For further perspective, the mean cross-validation error rates reported in a number of papers in the literature on the colon data are reproduced in Table 1.2. Slightly different pre-processing has been adopted in certain cases by different authors, leading to small differences in quoted results of those methods; most importantly, the number of training and testing samples used has a significant impact on the estimate of variance associated with the reported accuracy. Li et al. (2002) split the available 62 samples into 50 used in training their method and 12 test samples used to obtain unbiased estimates of their accuracy. They repeat this process for 100 random draws of the training set. Weston et al. (2003) use the same split but repeat their experiments for 500 random draws of the training set; hence their mean error rates should be comparable, even though their estimate of variance of error rates may not be directly comparable.

The JCFO typically identified around 25 features (genes) as important for the classification task. In contrast, using the RVM or the sparse probit algorithm with

Table 1.3 Most important genes for distinguishing between AML and ALL, as selected by the JCFO (Krishnapuram et al., 2004)

$\boldsymbol{\theta}_i$	**Gene Name**	**Gene Description**
1.14	MPO	myeloperoxidase
0.83	HOXA9	homeo box A9
0.81	APOC1	apolipoprotein C-I
0.77	PPGB	protective protein for beta-galactosidase (galactosialidosis)
0.70	NME4	non-metastatic cells 4, protein expressed in
0.67	PTGS2	prostaglandin-endoperoxide synthase 2
0.56	CD171	Human CD171 protein
0.51	NEK3	NIMA (never in mitosis gene a)-related kinase 3
0.46	CST3	cystatin C (amyloid angiopathy and cerebral hemorrhage)
0.42	EPB72	erythrocyte membrane protein band 7.2(stomatin)
0.42	DF	D component of complement (adipsin)
0.41	PTMA	prothymosin, alpha (gene sequence 28)
0.41	HSPA8	heat shock 70kDa protein 8
0.40	CYP2C18	cytochrome P450, subfamily IIC, polypeptide 18
0.35	CD33	CD33 antigen (gp67)
0.34	PRG1	proteoglycan 1, secretory granule
0.33	SERPING1	serine (or cysteine) proteinase inhibitor, clade G, member 1
0.32	FTH1	ferritin, heavy polypeptide 1
0.30	ALDH1A1	aldehyde dehydrogenase 1 family, member A1
0.29	LTC4S	leukotriene C4 synthase
0.27	MYBL1	v-myb myeloblastosis viral oncogene homolog (avian)-like 1
0.26	ITGA2B	integrin, alpha 2b (platelet glycoprotein IIb, antigen CD41B)
0.23	MACMARCKS	macrophage myristoylated alanine-rich C kinase substrate

no kernel to perform feature selection (as discussed in Section 1.2.3) resulted in classifiers based upon around 100 genes. More critically, for the RVM, each time the training set was changed in the LOOCV procedure, the genes identified as relevant for the discrimination changed significantly. In other words, this method was not stable to slight changes in the dataset, rendering its results less biologically trustworthy.

Biological relevance of identified genes

Table 1.3 lists the genes identified by the JCFO as being most important for making a diagnostic decision in distinguishing between AML and ALL. The reported values of $\boldsymbol{\theta}$ in these tables were obtained by taking the mean of the $\boldsymbol{\theta}$ obtained for each of the classifiers designed in the LOOCV. Almost all genes selected by the JCFO are of known relevance to the AML/ALL distinction. In particular, CST3, CD33, DF, HOXA9, LTC4S, PRG1, CTSD, and EPB72 were all determined both by the JCFO and in Golub et al. (1999) to be predictive of AML. In addition, the JCFO revealed MPO to be of paramount importance (though it was not uncovered in Golub et al. (1999)). MPO is known to occur in virtually all cells of the myeloid lineage and none of the lymphoid lineage; antibodies to MPO

Table 1.4 LOOCV accuracy of breast cancer diagnosis based on gene expression data for several classifiers (percent correct; a higher number is better; see text for descriptions of the datasets).

Classifier	**Duke ER**	**Duke LN**	**Lund**
SVM (linear kernel)	97.4	78.9	87.9
RVM (linear kernel)	94.7	92.1	88.5
RVM (no kernel)	89.5	81.6	96.5
Sparse probit regression (linear kernel)	97.4	89.5	86.2
Sparse probit regression (no kernel)	84.2	89.5	96.5
JCFO (linear kernel)	97.4	94.7	98.3

are used as clinical determinants of AML. Many others genes selected by the JCFO are known to play a role in myeloid/lymphoid differentiation, and a few novel genes have been identified as well. Similar results hold for the case of the colon data as well.

Breast cancer data

We also examined three different breast cancer datasets. The first was a Duke University study in which $m = 38$ breast tumors were classified based on estrogen receptor (ER) status. The second was a Duke University study in which the same $m = 38$ breast tumors were classified based on lymph node (LN) involvement status. The third was a set of $m = 58$ breast tissues collected by researchers at Lund University in Sweden. To accelerate the time required to complete the full LOOCV for all of the methods, and to give as much performance benefit as possible to the other classification methods that suffer more from the curse of dimensionality, the three breast cancer datasets were pared in advance to include only the 2000 most relevant genes as determined by FDR. Restricting the set of available genes in this way does not improve the accuracy of the JCFO because it is designed to perform feature selection as part of its optimization, but the smaller set of relevant initial features does improve the accuracy of the other methods.

In Table 1.4, we present a full leave-one-out cross-validation study for each of the three datasets to compare the accuracy of the diagnostic classification reported by the JCFO against that of the SVM, the RVM, and sparse probit regression. We consider only kernels that seemed to perform reasonably well in the previous tests (as shown in Table 1.1).

1.5 Discussion

An analysis of Table 1.1 and Table 1.4 suggests that for disease diagnosis tasks based on gene expression data, the JCFO provides superior classification accuracy to that of other popular methods like the SVM, RVM, and sparse probit regression. The statistical significance of this difference is unclear, though the fact that the JCFO performs well on a wide range of different tasks may be suggestive. Indeed, while we have not presented here results on other low-dimensional datasets widely used as benchmarks for classification, similar results have been obtained there as well. This may be understandable since the JCFO has two different and complementary kinds of regularization, one based on sparse choice of kernel basis functions and the other on feature selection.

Also, Table 1.2 seems to indicate that a variety of modern embedded feature selection methods such as the RVM, RFE, radius/margin bound minimization, and zero-norm minimization methods seem to perform comparably in terms of error rates of the learned classifier, and the differences between all of them are statistically small. However, the latter conclusion is somewhat tenuous due to the limited number of experimental results available for analysis; we are currently attempting to run those methods on more datasets in order to gather more statistical evidence. The relatively high classification accuracy obtained by all of these methods, including the JCFO, is indicative of the fact that gene expression data provides an excellent basis for distinguishing between disease classes.

Besides improved predictive accuracy in cross-validation experiments, the JCFO also provides the posterior probability of class membership as opposed to a hard classification decision as provided by the SVM, though this property is common to several other methods as well (like Gaussian processes). It tends to be more parsimonious in identifying only a small number of diagnostically relevant genes. We have also found that most of the genes have typically been implicated in earlier results published in the literature.

1.6 Availability of software

MATLAB implementations of sparse probit regression, RVM, JCFO, and forward feature selection with the SVM can be obtained by emailing the first author. The code is provided freely for non-commercial use.

MATLAB implementations of zero-norm minimization, RFE, and radius/margin bound minimization methods are publicly available in the Spider library from: `http://www.kyb.tuebingen.mpg.de/bs/people/spider/index.html`.

MATLAB implementations of the one-norm SVM software are publicly available from: `http://www.cs.wisc.edu/dmi/svm/`.

C++ implementations of the Generalized-LASSO, an algorithm similar to the RVM, are available from: `http://www.informatik.uni-bonn.de/~roth/GenLASSO/`.

S-Plus implementations of the RFE for penalized kernel logistic regression for academic research use were obtained by contacting Zhu and Hastie (2003) directly.

Acknowledgments

The authors would like to thank the editors for their gracious invitation to contribute a chapter to this volume. They would also like to gratefully acknowledge the assistance of Pallavi Pratapa in generating some of the JCFO classification results

and Mario Figueiredo for helpful comments regarding the text.

1.7 Appendix: Derivation for Q-function and E-step

As a first step, we see that the complete log-posterior on the learning parameters $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$, including the hidden variables $\boldsymbol{\tau}$, $\boldsymbol{\rho}$, and $\mathbf{z}$, is

$$\begin{aligned}\log P(\boldsymbol{\alpha}, \boldsymbol{\theta}|\mathbf{y}, \mathbf{z}, \boldsymbol{\rho}, \boldsymbol{\tau}) &\propto \log P(\mathbf{z}|\boldsymbol{\alpha}, \boldsymbol{\theta}) + \log P(\boldsymbol{\alpha}|\boldsymbol{\tau}) + \log P(\boldsymbol{\theta}|\boldsymbol{\rho}) \\ &\propto -\|\boldsymbol{H}\boldsymbol{\alpha} - \mathbf{z}\|^2 - \boldsymbol{\alpha}^T\boldsymbol{T}\boldsymbol{\alpha} - \boldsymbol{\theta}^T\boldsymbol{R}\boldsymbol{\theta} \qquad (1.17) \\ &\propto -\mathbf{z}^T\mathbf{z} - \boldsymbol{\alpha}^T\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{\alpha} - 2\mathbf{z}) - \boldsymbol{\alpha}^T\boldsymbol{T}\boldsymbol{\alpha} - \boldsymbol{\theta}^T\boldsymbol{R}\boldsymbol{\theta}\end{aligned}$$

where the matrix $\boldsymbol{T} = diag(\tau_1^{-1}, \tau_2^{-1}, \ldots, \tau_M^{-1})$ and $\boldsymbol{R} = diag(\rho_1^{-1}, \rho_2^{-1}, \ldots, \rho_d^{-1})$. Thus, the Q function is

$$\begin{aligned}Q\left(\boldsymbol{\alpha}, \boldsymbol{\theta} \,\middle|\, \hat{\boldsymbol{\theta}}^{(t)}, \hat{\boldsymbol{\alpha}}^{(t)}\right) = E\Big[&-\mathbf{z}^T\mathbf{z} - \boldsymbol{\alpha}^T\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{\alpha} - 2\mathbf{z}) \\ &- \boldsymbol{\alpha}^T\boldsymbol{T}\boldsymbol{\alpha} - \boldsymbol{\theta}^T\boldsymbol{R}\boldsymbol{\theta} \,\Big|\, \mathbf{y}, \hat{\boldsymbol{\alpha}}^{(t)}, \hat{\boldsymbol{\theta}}^{(t)}\Big] \qquad (1.18)\end{aligned}$$

Since we seek to maximize the Q function w.r.t. $\boldsymbol{\alpha}$ in the EM algorithm, terms like $E\left[-\mathbf{z}^T\mathbf{z} \,\middle|\, \mathbf{y}, \hat{\boldsymbol{\theta}}^{(t)}, \hat{\boldsymbol{\alpha}}^{(t)}\right]$ that do not involve $\boldsymbol{\alpha}$ or $\boldsymbol{\theta}$ can be effectively ignored in the M-step, and thus are irrelevant in the E-step as well. Therefore, the Q function simplifies to

$$\begin{aligned}Q\left(\boldsymbol{\alpha}, \boldsymbol{\theta} \,\middle|\, \hat{\boldsymbol{\theta}}^{(t)}, \hat{\boldsymbol{\alpha}}^{(t)}\right) &= -\boldsymbol{\alpha}^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^T\boldsymbol{H}^T E\left[\mathbf{z} \,\middle|\, \mathbf{y}, \hat{\boldsymbol{\theta}}^{(t)}, \hat{\boldsymbol{\alpha}}^{(t)}\right] \\ &- \boldsymbol{\alpha}^T E\left[\boldsymbol{T} \,\middle|\, \mathbf{y}, \hat{\boldsymbol{\theta}}^{(t)}, \hat{\boldsymbol{\alpha}}^{(t)}\right]\boldsymbol{\alpha} - \boldsymbol{\theta}^T E\left[\boldsymbol{R} \,\middle|\, \mathbf{y}, \hat{\boldsymbol{\theta}}^{(t)}, \hat{\boldsymbol{\alpha}}^{(t)}\right]\boldsymbol{\theta} \qquad (1.19)\end{aligned}$$

The E-step thus simplifies to computing the expectations associated with each of these terms. Fortunately, each of these computations can be expressed in closed form, as shown below.

As for the term associated with the expectation of $\mathbf{z}$, we have

$$v_i = E\left[z^{(i)} \,\middle|\, \mathbf{y}, \hat{\boldsymbol{\alpha}}^{(t)}, \hat{\boldsymbol{\theta}}^{(t)}\right] = \begin{cases}\boldsymbol{h}^T(\mathbf{x}^{(i)})\hat{\boldsymbol{\alpha}}^{(t)} + \frac{N\left(\boldsymbol{h}^T(\mathbf{x}^{(i)})\hat{\boldsymbol{\alpha}}^{(t)} \middle| 0,1\right)}{1-\Phi\left(-\boldsymbol{h}^T(\mathbf{x}^{(i)})\hat{\boldsymbol{\alpha}}^{(t)}\right)}, & \text{if } y^{(i)} = 1 \\ \boldsymbol{h}^T(\mathbf{x}^{(i)})\hat{\boldsymbol{\alpha}}^{(t)} - \frac{N\left(\boldsymbol{h}^T(\mathbf{x}^{(i)})\hat{\boldsymbol{\alpha}}^{(t)} \middle| 0,1\right)}{\Phi\left(-\boldsymbol{h}^T(\mathbf{x}^{(i)})\hat{\boldsymbol{\alpha}}^{(t)}\right)}, & \text{if } y^{(i)} = 0\end{cases} \qquad (1.20)$$

which follows from the observation that $z^{(i)}$ is distributed as a Gaussian with mean $\boldsymbol{h}^T(\mathbf{x}^{(i)})\hat{\boldsymbol{\alpha}}^{(t)}$, but left-truncated at zero if $y^{(i)} = 1$, and right-truncated at zero if $y^{(i)} = 0$. This expression can be simplified further to remove the case statement, as shown earlier in (1.12).

After some further algebraic manipulations, it can be shown that for the expectation of τ_i^{-1} is given by

$$\omega_i = E\left[\tau_i^{-1} \;\middle|\; \mathbf{y}, \hat{\boldsymbol{\alpha}}_i^{(t)}, \gamma_1\right] = \frac{\int\limits_0^\infty \tau_i^{-1} P(\tau_i|\gamma_1) P\left(\hat{\boldsymbol{\alpha}}_i^{(t)} \;\middle|\; \tau_i\right) d\tau_i}{\int\limits_0^\infty P(\tau_i|\gamma_1) P\left(\hat{\boldsymbol{\alpha}}_i^{(t)} \;\middle|\; \tau_i\right) d\tau_i} = \gamma_1 \left|\hat{\boldsymbol{\alpha}}_i^{(t)}\right|^{-1} \tag{1.21}$$

The last term in the E-step computation is associated with the expectation of $\boldsymbol{R}$, and a manipulation similar to that above yields the following:

$$\delta_i = E\left[\rho_i^{-1} \;\middle|\; \mathbf{y}, \hat{\boldsymbol{\alpha}}^{(t)}, \hat{\boldsymbol{\theta}}_i^{(t)}, \gamma_2\right] = \gamma_2 \left(\hat{\boldsymbol{\theta}}_i^{(t)}\right)^{-1} \tag{1.22}$$

References

J. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88:669–679, 1993.

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. *Proceedings of the National Academy of Science*, 96:6745–6750, 1999.

C. Ambroise and G. J. McLachlan. Selection bias in gene extraction on the basis of microarray gene expression data. *Proceedings of the National Academy of Science*, 99(10):6562–6566, 2002.

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)*, Tokyo, 2000. Universal Academy Press.

A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, Cambridge, UK, 2000.

M. A. T. Figueiredo. Adaptive sparseness for supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, 2003.

M. A. T. Figueiredo and A. K. Jain. Bayesian learning of sparse classifiers. In *2001 Conference on Computer Vision and Pattern Recognition (CVPR 2001).* IEEE Press, December 2001.

J. Friedman, T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. Discussion of "consistency in boosting". *Annals of Statistics*, 2004.

G. Fung and O. Mangasarian. A feature selection Newton method for support vector machine classification. Technical report, Data Mining Institute, University of Wisconsin (Madison), September 2002.

T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

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. Caliguiri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature: Medicine*, 7(6):673–679, 2001.

R. Kohavi and G. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

B. Krishnapuram, L. Carin, and A. Hartemink. Joint classifier and feature optimization for comprehensive cancer diagnosis using gene expression data. *Journal of Computational Biology*, 2004.

B. Krishnapuram, M. A. T. Figueiredo, L. Carin, and A. Hartemink. An EM algorithm for joint feature selection and classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page (submitted), 2003a.

B. Krishnapuram, M. A. T. Figueiredo, L. Carin, and A. Hartemink. Learning sparse Bayesian classifiers: Multi-class formulation, fast algorithms, and generalization bounds. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, page (submitted), 2003b.

Y. Li, C. Campbell, and M. Tipping. Bayesian automatic relevance determination algorithms for classifying gene expression data. *Bioinformatics*, 18:1332–1339, 2002.

D. Lockhart, H. Dong, M. Byrne, M. Follettie, M. Gallo, M. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, and E. Brown. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, 14 (12):1675–1680, 1996.

O. Mangasarian. Arbitrary-norm separating plane. *Operations Research Letters*, 24(1-2):15–23, 1999.

A. Rakotomamonjy. Variable selection using SVM based criteria. *Journal of Machine Learning Research*, 3:1357–1370, 2003.

S. Ramaswamy et al. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Science*, 98:15149–15154, 2001.

S. Rosset, J. Zhu, and T. Hastie. Margin maximizing loss functions. In *Advances in Neural Information Processing Systems (NIPS) 15*. MIT Press, 2003.

V. Roth. The generalized LASSO. *IEEE Transactions on Neural Networks*, 2003.

M. Schena, D. Shalon, R.W. Davis, and P.O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270:

467–70, 1995.

M. Seeger. Bayesian model selection for support vector machines, Gaussian processes, and other kernel classifiers. In *Advances in Neural Information Processing Systems (NIPS) 12*. MIT Press, 2000.

R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society (B)*, 58:267–288, 1996.

M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995. ISBN 0-387-94559-8.

V. E. Velculescu, L. Zhang, B. Vogelstein, and K. W. Kinzler. Serial analysis of gene expression. *Science*, 270:484–487, 1995.

J. Weston, A. Elisseff, B. Schölkopf, and M. Tipping. Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3: 1439–1461, 2003.

J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. In *Advances in Neural Information Processing Systems (NIPS) 12*. MIT Press, 2000.

C. K. I. Williams and D. Barber. Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(20), 1998.

M. Xiong, W. Li, J. Zhao, L. Jin, and E. Boerwinkle. Feature (gene) selection in gene expression-based tumor classification. *Molecular Genetics and Metabolism*, 73(3):239–247, 2001.

J. Zhu and T. Hastie. Classification of gene microarrays by penalized logistic regression. *Biostatistics*, 2003.