Expression Arrays and the $p \gg n$ Problem

Trevor Hastie* Robert Tibshirani †

20 November, 2003

Abstract

Gene expression arrays typically have 50 to 100 samples and 5,000 to 20,000 variables (genes). There have been many attempts to adapt statistical models for regression and classification to these data, and in many cases these attempts have challenged the computational resources. In this article we expose a class of techniques based on quadratic regularization of linear models, including regularized (ridge) regression, logistic and multinomial regression, linear and mixture discriminant analysis, the Cox model and neural network. For all of these models, we show that dramatic computational saving are possible over naive implementations, using standard transformations in numerical linear algebra.

Keywords: quadratic regularization, euclidean methods, QR decomposition

1 Introduction

Suppose we have an expression array $\mathbf{X}$ consisting of n samples and p genes. In keeping with statistical practice the dimensions of $\mathbf{X}$ is n rows by p columns; hence its transpose $\mathbf{X}^T$ gives the traditional biologists view of the vertical skinny matrix where the ith column is a microarray sample x_i. Expression arrays have orders of magnitude more genes than samples, hence $p \gg n$. We often have accompanying data that characterize the samples, such as cancer class, biological species, survival time, or other quantitative measurements. We will denote by y_i such a description for for sample i. A common statistical task is to build a prediction model that uses the vector of expression values x for a sample as the input to predict the output value y.

In this article we discuss the use of standard statistical models in this context, such as the linear regression model, logistic regression and the Cox model, and linear discriminant analysis, to name a few. These models cannot be used "out of the box", since the standard fitting algorithms all require $p < n$; in fact the usual rule of thumb is that there be five or ten times as many samples as

*(corresponding author) Departments of Statistics, and Health, Research & Policy, Sequoia Hall, Stanford University, CA 94305. hastie@stanford.edu

†Departments of Health, Research & Policy, and Statistics, Stanford University, tibs@stanford.edu

variables. But here we consider situations with n around 50 or 100, while p typically varies between 1,000 and 20,000.

There are several ways to overcome this dilemma. These include

- dramatically reducing the number of genes to bring down p. This can be done by univariate screening of the genes, using, for example, $t-$tests [Tusher et al., 2001, e.g.].
- Use a constrained method for fitting the model, such as naive Bayes, that does not fit all p parameters freely [Tibshirani et al., 2003].
- Use a standard fitting method along with regularization.

In this article we focus on the third of these approaches, and in particular quadratic regularization. We also show how the computations can be dramatically reduced for a large class of quadratically regularized linear models.

2 Linear Regression and Quadratic Regularization

Consider the usual linear regression model $y_i = x_i^T\beta + \epsilon_i$ and its associated least-squares fitting criterion

$$\min_{\beta} \sum_{i=1}^{n} (y_i - x_i^T\beta)^2 \tag{1}$$

The textbook solution $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ does not work when $p > n$, since in this case the $p \times p$ matrix $\mathbf{X}^T\mathbf{X}$ has rank at most n, and is hence singular and cannot be inverted. A more accurate description is that the "normal equations" that lead to this expression, $\mathbf{X}^T\mathbf{X}\beta = \mathbf{X}^T\mathbf{y}$ do not have a unique solution for β, and infinitely many solutions are possible. Moreover they all lead to a perfect fit; perfect on the training data, but unlikely to be of much use for future predictions.

The "ridge regression" solution to this dilemma is to modify (1) by adding a quadratic penalty

$$\min_{\beta} \sum_{i=1}^{n} (y_i - x_i^T\beta)^2 + \lambda\beta^T\beta \tag{2}$$

for some $\lambda > 0$ This gives

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}, \tag{3}$$

and the problem has been fixed since now $\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}$ is invertible. The effect of this penalty is to constrain the size of the coefficients by shrinking them toward zero. More subtle effects are that coefficients of correlated variables (genes, of which there are many) are shrunk toward each other as well as toward zero.

Remarks:

- In our model above, we have ignored the intercept for notational simplicity. Typically an intercept is included, and hence the model is $f(x) = \beta_0 + x^T\beta$, but we do not penalize β_0 when doing the fitting. In this particular case we can rather work with centered variables (from each of the genes subtract its mean), which implies that the unpenalized estimate $\hat{\beta}_0$ is the mean of the y_i.

- Often in ridge regression, the predictor variables are measured in different units. To make the penalty meaningful, it is typically recommended that the variables be standardized first to have unit sample variance. In the case of expression arrays, the variables (genes) are all measured in the same units, so this standardization is optional.

- The tuning parameter λ controls the amount of shrinkage, and has to be selected by some external means. We demonstrate the use of K-fold cross-validation for this purpose in the examples later on.

It appears the ridge solution (3) is very expensive to compute, since it requires the inversion of a $p \times p$ matrix (which takes $O(p^3)$ operations). Here we demonstrate a computationally efficient solution to this problem.

Let $\mathbf{X}^T = \mathbf{Q}_1\mathbf{R}^T$ be a Q-R decomposition of the transpose of $\mathbf{X}$; that is, $\mathbf{Q}_1$ is $p \times n$ and has n orthonormal columns ($\mathbf{Q}_1^T\mathbf{Q}_1 = \mathbf{I}_n$), and $\mathbf{R}_{n\times n}$ is a square $n \times n$ matrix with rank at most n. Plugging this into (3), and after some careful linear algebra, we find that

$$\hat{\beta} = \mathbf{Q}_1(\mathbf{R}^T\mathbf{R} + \lambda\mathbf{I})^{-1}\mathbf{R}^T\mathbf{y}. \tag{4}$$

Comparing with (3), we see that (4) is the ridge-regression coefficient using the much smaller $n \times n$ regression matrix $\mathbf{R}$, pre-multiplied by $\mathbf{Q}_1$. In other words, we can solve the ridge regression problem involving p variables, by

- reducing the p variables to $n \ll p$ variables via the Q-R decomposition in $O(pn^2)$ operations;

- solving the n dimensional ridge regression problem in $O(n^3)$ operations;

- transforming the solution back to to p dimensions in $O(np)$ operations.

Thus the computational cost is reduced from $O(p^3)$ to $O(pn^2)$ when $p > n$.

3 Linear Models and Quadratic Penalties

There are many other situations where linear models are used. Examples include logistic and multinomial regression, linear and mixture discriminant analysis, the Cox model, linear support-vector machines, and neural networks. We discuss some of these in more detail later in the paper. All these models produce a function $f(x)$ that involve x via one or more linear function. They are typically used in situations where $p < n$, and are fit my minimizing some loss function

$\sum_{i=1}^n L(y_i, f(x_i))$ over the data. Here L can be squared error, negative log-likelihood, negative partial log-likelihood, etc. All suffer in a similar fashion when $p \gg n$, and all can be *fixed* by quadratic regularization:

$$\min_{\beta_0,\beta} \sum_{i=1}^n L(y_i, \beta_0 + x_i^T\beta) + \lambda\beta^T\beta. \tag{5}$$

For the case of more than one set of linear coefficients, we can simply add more quadratic penalty terms (e.g. for multinomial regression and neural network models).

We now show that the Q-R trick used for ridge regression works in exactly the same way for all these problems.

Let $\mathbf{X}^T = \mathbf{Q}_1\mathbf{R}^T$ be a Q-R decomposition of $\mathbf{X}^T$ as before, and denote by $r_i,\ i = 1, \ldots, n$ the n rows of $\mathbf{R}$, each an n-vector of predictors. Then we have the following simple theorem

Theorem 1 *Consider the pair of optimization problems:*

$$(\hat\beta_0, \hat\beta) = \operatorname*{argmin}_{\beta_0,\beta\in\mathbb{R}^p} \sum_{i=1}^n L(y_i, \beta_0 + x_i^T\beta) + \lambda\beta^T\beta; \tag{6}$$

$$(\hat\theta_0, \hat\theta) = \operatorname*{argmin}_{\theta_0,\theta\in\mathbb{R}^n} \sum_{i=1}^n L(y_i, \theta_0 + r_i^T\theta) + \lambda\theta^T\theta. \tag{7}$$

Then $\hat\beta_0 = \hat\theta_0$, *and* $\hat\beta = \mathbf{Q}_1\hat\theta$.

The theorem says that we can simply replace the p vectors x_i by the n-vectors r_i, and perform our penalized fit as before, except with much fewer predictors. The n-vector solution $\hat\theta$ is then transformed back to the p-vector solution via a simple matrix multiplication.

Proof

Let $\mathbf{Q}_2$ be $p\times(p-n)$ and span the complementary subspace in $\mathbb{R}^p$ to $\mathbf{Q}_1$. Then $\mathbf{Q} = (\mathbf{Q}_1 : \mathbf{Q}_2)$ is a $p \times p$ orthonormal matrix. Let $x_i^* = \mathbf{Q}^T x_i$ and $\beta^* = \mathbf{Q}^T\beta$. Then

- ${x_i^*}^T\beta^* = x_i^T\mathbf{Q}\mathbf{Q}^T\beta = x_i^T\beta$, and
- ${\beta^*}^T\beta^* = \beta^T\mathbf{Q}\mathbf{Q}^T\beta = \beta^T\beta$.

Hence the criterion (6) is invariant under orthogonal transformations. In other words, the criterion with x_i, β_0 and β is identical to that with x_i^*, β_0 and β^*. Hence there is a one-one mapping between the location of their minima, so we can focus on β^* rather than β. But ${x_i^*}^T\beta^* = r_i^T\beta_1^*$, where β_1^* consists of the first n elements of β^*. This follows from the Q-R decomposition of $\mathbf{X}^T$. Hence the loss part of the criterion (6) involves β_0 and β_1^*. We can similarly factor the quadratic penalty into two terms $\lambda{\beta_1^*}^T\beta_1^* + \lambda{\beta_2^*}^T\beta_2^*$. Hence the criterion in (6) decouples into a sum of two independent parts,

$$\left[\sum_{i=1}^n L(y_i, \beta_0 + r_i^T\beta_1^*) + \lambda{\beta_1^*}^T\beta_1^*\right] + \left[\lambda{\beta_2^*}^T\beta_2^*\right], \tag{8}$$

which we can minimize separately. The second part is minimized at $\beta_2^* = 0$, and the result follows by noting that the first part is identical to the criterion in (7) with $\theta_0 = \beta_0$ and $\theta = \beta_1^*$.

□

Theorem 1 is not deep, but its implications allow a great reduction in computations for all the models described above.

In many situations, such as when the loss function is based on a log-likelihood, we use the criterion itself and its derivatives as the basis for inference. Examples are profile likelihoods, score tests based on the first derivatives, and (asymptotic) variances of the parameter estimates based on the information matrix (second derivatives). We now see that we can obtain these p-dimensional functions from the corresponding n-dimensional versions.

Corollary 2

$$\sum_{i=1}^{n} L(y_i, \beta_0 + x_i^T\beta) = \sum_{i=1}^{n} L(y_i, \beta_0 + r_i^T\theta) \tag{9}$$

with $\theta = \mathbf{Q}_1^T\beta$. If the loss function L in (6) is differentiable, then

$$\frac{\partial L(\beta)}{\partial \beta} = \mathbf{Q}_1 \frac{\partial L(\theta)}{\partial \theta}; \tag{10}$$

$$\frac{\partial^2 L(\beta)}{\partial \beta \partial \beta^T} = \mathbf{Q}_1 \frac{\partial^2 L(\theta)}{\partial \theta \partial \theta^T} \mathbf{Q}_1^T, \tag{11}$$

with the partial derivatives on the right hand side evaluated at $\theta = \mathbf{Q}_1^T\beta$.

Notes:

- In (10) and (11), $L(\beta) = \sum_{i=1}^n L(y_i, \beta_0 + x_i^T\beta)$ and $L(\theta) = \sum_{i=1}^n L(y_i, \beta_0 + r_i^T\theta)$.
- These equations hold at all values of the parameters, not just at the solutions.
- Obvious modifications apply if we include the penalty in these derivatives.

Proof

Equation (9) follows immediately from the identity $\mathbf{X} = \mathbf{R}\mathbf{Q}_1^T$, and the fact that x_i^T and r_i^T are the ith rows of $\mathbf{X}$ and $\mathbf{R}$. The derivatives (10) and (11) are simple applications of the chain rule to (9).

□

Q-R decompositions are standard linear algebra tools, and require $O(pn^2)$ computations. They amounts to rotating the observed data in R^p to a new coordinate system, in which the data have nonzero coordinates on only the first n dimensions. As such they are not unique, since any further rotation in the n-dimensional subspace would remain in this form. Two standard Q-R decompositions are

- Gram-Schmidt orthogonalization, in which $\mathbf{R}$ is upper triangular;
- The singular value decomposition (SVD) $\mathbf{X}^T = \mathbf{U}\mathbf{D}\mathbf{V}^T$, where $\mathbf{U}$ is $p \times n$ orthogonal, $\mathbf{V}$ is $n \times n$ orthogonal, and $\mathbf{D}$ is $n \times n$ diagonal, non-negative, with elements in non-increasing order on the diagonal. Here we would identify $\mathbf{U}$ with $\mathbf{Q}_1$ and $\mathbf{R} = \mathbf{D}\mathbf{V}^T$.

4 Examples of Regularized Linear Models

In this section we document a large class of linear models where quadratic regularization can be used in a similar manner, and the same computational trick of using r_i rather than x_i can be used.

Logistic Regression

Logistic regression is the traditional linear model used when the response variable is binary. The class conditional probability is represented by

$$\Pr(y = 1|x) = \frac{e^{\beta_0 + x^T\beta}}{1 + e^{\beta_0 + x^T\beta}} \tag{12}$$

The parameters are typically fit by maximizing the binomial log-likelihood

$$\sum_{i=1}^{n} \{y_i \log p_i + (1 - y_i) \log(1 - p_i)\}, \tag{13}$$

where we have used the shorthand notation $p_i = \Pr(y = 1|x_i)$.

The maximum likelihood solution is obtained by setting the gradient of the log-likelihood (score equations) to zero:

$$\begin{array}{rcl} \mathbf{1}^T(\mathbf{y} - \mathbf{p}) & = & 0 \\ \mathbf{X}^T(\mathbf{y} - \mathbf{p}) & = & 0, \end{array} \tag{14}$$

where $\mathbf{p}$ is the n-vector of probabilities p_i. If $p > n - 1$, maximum-likelihood estimation would fail for the same reasons OLS did for ordinary regression; there are $p + 1 > n$ parameters, and $\mathbf{X}$ has rank at most n, so there are at most n linearly independent score equations among the $p + 1$ in (14).

This problem can again be fixed by using a quadratic penalty. We maximize instead the penalized log-likelihood

$$\sum_{i=1}^{n} y_i \log p_i + (1 - y_i) \log(1 - p_i) - \lambda\beta^T\beta, \tag{15}$$

leading to the ridged score equations

$$\begin{array}{rcl} \mathbf{1}^T(\mathbf{y} - \mathbf{p}) & = & 0 \\ \mathbf{X}^T(\mathbf{y} - \mathbf{p}) - \lambda\mathbf{I} & = & 0, \end{array} \tag{16}$$

and a unique solution is obtained.

Remarks:

- When $p \gg n$, this usually means the two classes are perfectly separable by a linear affine boundary (unless degeneracies occur, such as two points from different classes have exactly the same value for x). Even for $p < n$, the data can be linearly separable. Maximum likelihood estimates for logistic regression with perfectly separable data are undefined (parameters march off to infinity); the regularization fixes this, and provides a unique solution in either of the above cases.
- In the separable case above, as $\lambda \downarrow 0$, the sequence of solutions $\hat{\beta}(\lambda)$ (suitably normalized) converge to the optimal separating hyperplane; i.e. the same solution as the support-vector machine [Rosset et al., 2003]; see below.
- The binary logistic regression model generalizes naturally to the multi-logit model. The regularized solution generalizes automatically, as does the limit as $\lambda \downarrow 0$.

Our theorem tells us that we can fit instead a regularized logistic regression using the r_i as observations, instead of the x_i.

Generalized Linear Models

Linear regression by least squares fitting and logistic regression are part of the class of *generalized linear models*. For this class we assume the regression function $\mathrm{E}(y|x) = \mu(x)$, and that $\mu(x)$ is related to the inputs via the monotone *link* function g: $g(\mu(x)) = f(x) = \beta_0 + x^T\beta$. The log-linear model for responses y_i that are counts is another important member of this class. These would all be fit by regularized maximum likelihood if $p \gg n$.

The Cox Proportional Hazards Model

This model is used when the response is survival time (possibly censored). The hazard function is modeled as $\lambda(t|x) = \lambda_0(t)e^{x^T\beta}$. Here there is no intercept, since it is absorbed into the baseline hazard $\lambda_0(t)$. A *partial likelihood* [Cox, 1972] is typically used for inference, regularized if $p \gg n$.

Multiple Logistic Regression

This model generalizes the logistic regression model when there are $K > 2$ classes. The model has the form

$$Pr(y = j|x) = \frac{e^{\beta_{0j}+\beta_j^T x}}{\sum_{\ell=1}^{K} e^{\beta_{0\ell}+\beta_\ell^T x}} \tag{17}$$

When $p > n$, this model would be fit by maximum penalized log-likelihood, based on the multinomial distribution

$$\max_{\{\beta_{0j},\beta_j\}_{j=1}^{K}} \sum_{i=1}^{n} \log \Pr(y_i|x_i) - \lambda \sum_{j=1}^{K} \beta_j^T \beta_j. \tag{18}$$

There is some redundancy in the representation (17), since we can add a constant c_m to all the class coefficients for any variable x_m, and the probabilities do not change. Typically in logistic regression, this redundancy is overcome by arbitrarily setting the coefficients for one class to zero (typically class K). Here this is not necessary, because of the regularization penalty; the c_m are chosen automatically to minimize the L_2 norm of the set of coefficients. Since the constant terms β_{0j} are not penalized, this redundancy persists, but we still choose the minimum-norm solution. This model is discussed in more detail in Zhu and Hastie [2003]. We illustrate this model on an example in Section 5. Even though there are multiple coefficient vectors β_j, it is easy to see that we can once again fit the multinomial model using the reduced predictors r_i.

Neural Networks

Single layer neural networks have hidden units $z_m = \sigma(\beta_{0m} + \beta_m^T x)$ that are linear functions of the inputs, and then another linear/logistic/multilogit model that takes the z_m as inputs. Here there are two layers of linear models, and both can benefit from regularization. Once again, quadratic penalties on the β_m allow us to reparameterize the first layer in terms of the r_i rather than the x_i.

Linear Support Vector Machines

The support vector machine (SVM) [Vapnik, 1996] for two-class classification is a popular method for classification. This model fits an optimal separating hyperplane between the data points in the two classes, with built in slack variables and regularization to handle the case when the data cannot be linearly separated. The problem is usually posed as an application in convex optimization. With y_i coded as $\{-1, +1\}$, it can be shown [Hastie et al., 2001] that the problem

$$\min_{\beta_0,\beta} \sum_{i=1}^{n} (y_i - \beta_0 - x_i^T \beta)_+ + \lambda \beta^T \beta \tag{19}$$

is an equivalent formulation of this optimization problem, and is of the form (5). In (19) we have used the *hinge loss* function for an SVM model, where the "+" denotes *positive part*.

4.1 The Kernel Trick

Users of SVM technology will recognize that our computational device must amount to some version of the "kernel" trick, which has been applied in many of the situations listed above. For linear models, the kernel trick amounts to a different reparametrization of the data, also from p down to n dimensions. For example, the vector of fitted values (ignoring the intercept) is represented as

$$\begin{aligned} \mathbf{f} &= \mathbf{X}\beta \\ &= \mathbf{X}\mathbf{X}^T\alpha \end{aligned} \tag{20}$$

$$= \quad \mathbf{K}\alpha.$$

The *gram* matrix $\mathbf{K} = \mathbf{X}\mathbf{X}^T$ represents the $n \times n$ inner-products between all pairs input vectors in the data. The new input variables are the n kernel basis functions $K(x, x_i) = x^T x_i,\ i = 1, \ldots, n$.

From (20) it is clear that the parametrization recognizes that $\beta = \mathbf{X}^T\alpha$ is in the row space of $\mathbf{X}$, just a different parametrization of our $\beta = \mathbf{Q}_1\theta$. However, with the parametrization (20), the general criterion in (5) becomes

$$\min_{\beta_0,\alpha} \sum_{i=1}^{n} L(y_i, \beta_0 + k_i^T\alpha) + \lambda\alpha^T\mathbf{K}\alpha, \tag{21}$$

where k_i is the ith row of $\mathbf{K}$. Hence our reparametrization r_i includes in addition an orthogonalization which diagonalizes the penalty in (21), leaving the problem in the same form as the original diagonal penalty problem.

The kernel trick allows for more flexible modeling, and is usually approached in the reverse order. A positive-definite kernel $K(x, x')$ generates a set of n basis functions $K(x, x_i)$, and hence a regression model $f(x) = \beta_0 + \sum_{i=1}^{n} K(x, x_i)\alpha_i$. A popular example of such a kernel is the radial basis function

$$K(x, x') = e^{-\gamma||x-x'||^2}, \tag{22}$$

or Gaussian bump function. The optimization problem is exactly the same as in (21). What is often not appreciated is that the roughness penalty on this space is induced by the kernel as well, as is evidenced in (21).

The nature of this penalty is best understood in terms of the implicit, often infinite dimensional *feature space*, for which $K(x, x')$ is assumed to compute inner-products as in (20). More precisely, a kernel $K(x, x')$ generates a *Reproducing Kernel Hilbert Space* of functions: $f(x) \in \text{span}\{K(x, x'), x' \in \mathbb{R}\}$, an infinite space with a built in regularization norm. See Hastie et al. [2001] for details.

Note that even for these more general kernel problems, we can reduce the parametrization to a diagonal form. Let $\mathbf{K} = \mathbf{R}\mathbf{R}^T$ be the Choleski decomposition of $\mathbf{K}$. Then with $k_i^* = \mathbf{R}^{-1}k_i$, and $\theta = \mathbf{R}^T\alpha$, a nonsingular transformation, the penalty is diagonal. Note that this transformation reduces the linear gram matrix $\mathbf{K} = \mathbf{X}\mathbf{X}^T$ exactly to the parametrization r_i proposed in this paper.

4.2 Euclidean Distance Methods

A number of multivariate methods rely on the euclidean distances between pairs of observations. K-means clustering and nearest-neighbor classification methods are two popular examples. It is easy to see that for such methods, we can also work with the r_i rather than the original x_i, since such methods are rotationally invariant.

- With K-means clustering, we would run the entire algorithm in the reduced space. The subclass means $\bar{r}_m$ could then be transformed back into the original space $\bar{x}_m = \mathbf{Q}_1\bar{r}_m$. The cluster assignments are unchanged.

- With k-nearest-neighbor classification we would drop the query point x into the n-dimensional subspace, $r = \mathbf{Q}_1^T x$, and then classify according to the labels of the closest k r_i.

The same is true for hierarchical clustering, even when the correlation "distance" is used.

5 Examples on Data

We illustrate some of these methods on a large cancer expression data set[Ramaswamy et al., 2001]. There are 144 training tumor samples and 54 test tumor samples, spanning 14 common tumor classes that account for 80% of new cancer diagnoses in the U.S. Among these 54 test samples, 8 are metastatic samples. There are 16,063 genes for each sample. Hence $p = 16,063$ and $n = 144$, in our terminology. We denote the number of classes by $K = 14$.

5.1 Multinomial Regression

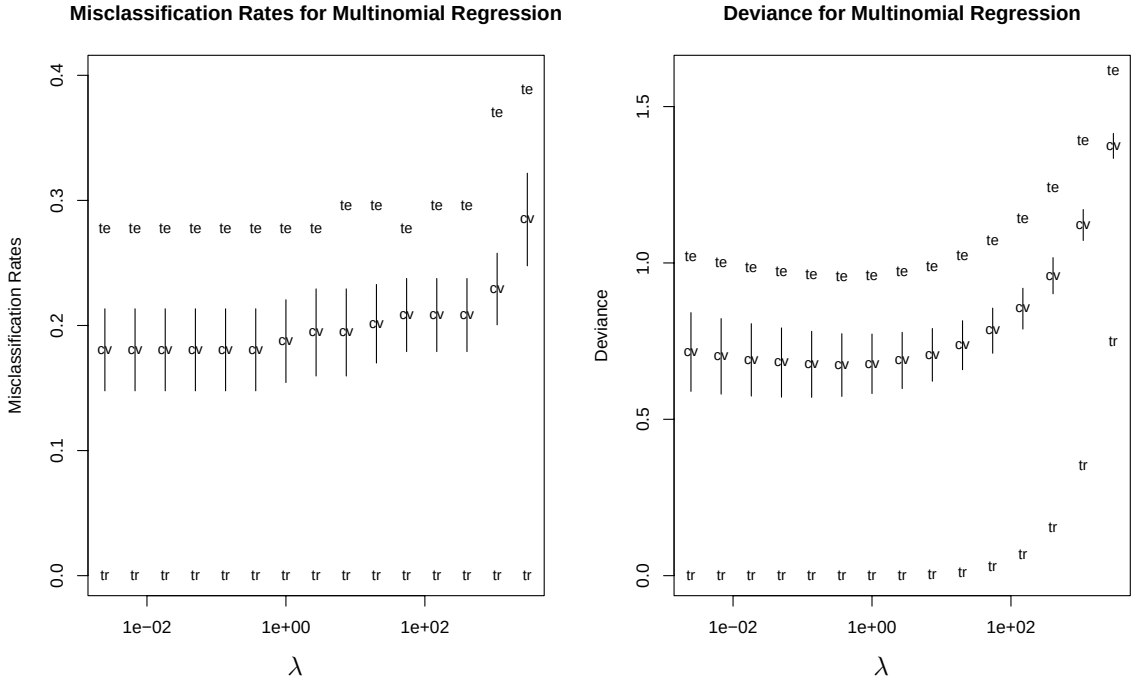

Figure 1: *Misclassification rates and deviance (2× negative log-likelihood) for the 14-class cancer data. The labels indicate training data (tr), test data (te), and 8-fold cross-validation (cv). The minimum number of test errors was 15.*

Figure 1[left panel] shows the results of fitting the multinomial model (17) for different values of λ. Shown are the training error, test error, as well as 8-fold balanced cross-validation[1]. In the right panel we show the deviance or negative log-likelihood of the fitted models. The deviance measures the fit of the model in terms of the fitted probabilities, and is smoother than misclassification error rates. We see that a good choice of λ is about 1 for these data; larger than that and the error rates (CV and test) start to increase.

These error rates might seem fairly high (0.27 or 15 misclassified test observations at best). For these data the null error rate is 0.89 (assign all test observations to the dominant class), which is indicative of the difficulty of multi-class classification. When this model is combined with redundant feature elimination [Zhu and Hastie, 2003], the test error rate drops to 0.18 (9 misclassifications).

The multinomial model not only learns the classification, but also provides estimates for the probabilities for each class. These can be used to assign a strength to the classifications. For example, one of the misclassified test observations had a probability estimate of 0.46 for the incorrect class, and 0.40 for the correct class; such a close call with 14 classes competing might well be assigned to the unsure category. For 6 of the 15 misclassified test observations, the true class had the second highest probability score.

5.2 Regularized Linear Discriminant Analysis

Here we demonstrate another multi-class classification model, that does not appear to be in the class (5). The linear discriminant analysis model is based on an assumption that the input features have a multivariate Gaussian distribution in each of the classes, with different mean vectors μ_k, but a common covariance matrix Σ. It is then easy to show that the log posterior probability for class k is given (up to a factor independent of class) by the *discriminant function*

$$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log \pi_k, \tag{23}$$

where π_k is the *prior probability* or background relative frequency of class k. Note that $\delta_k(x)$ is linear in x. We then classify to the class with the largest $\delta_k(x)$. In practice estimates

$$\begin{aligned} \hat{\pi}_k &= \frac{n_k}{n} \\ \hat{\mu}_k &= \frac{1}{n_k}\sum_{y_i=k} x_i \\ \hat{\Sigma} &= \frac{1}{n-k}\sum_{k=1}^{K}\sum_{y_i=k}(x_i-\hat{\mu}_k)(x_i-\hat{\mu}_k)^T \end{aligned} \tag{24}$$

[1]By balanced we mean the 14 cancer classes were represented equally in each of the folds; 8 folds were used to accommodate this balance, since the class sizes in the training set were multiples of 8

are plugged into (23) giving the estimated discriminant functions $\hat{\delta}_k(x)$. However, $\hat{\Sigma}$ is $p \times p$ and has rank at most $n - K$, and so its inverse in (23) is undefined. *Regularized discriminant analysis* or RDA [Friedman, 1989, Hastie et al., 2001] fixes this by replacing $\hat{\Sigma}$ with $\hat{\Sigma}(\lambda) = \hat{\Sigma} + \lambda \mathbf{I}$, which is nonsingular if $\lambda > 0$.

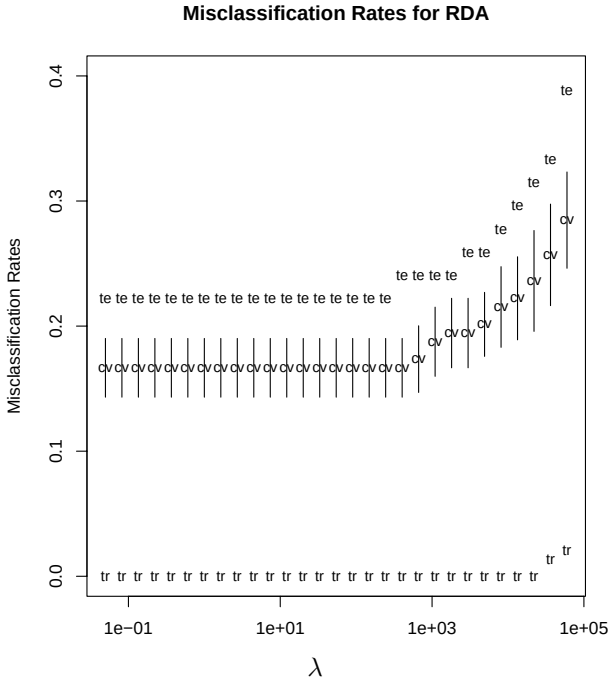

Figure 2: *Misclassification rates for the 14-class cancer data. The labels indicate training data (tr), test data (te), and 8-fold cross-validation (cv). The minimum number of test errors is 12.*

Now (23) and (24) do not appear to be covered by (5) and Theorem 1.[2]. However, it is simple to show directly that (23) and its regularized version are invariant under a coordinate rotation. Hence we can once again use the Q-R construction and replace the training x_i by their corresponding r_i, and fit the RDA model in the lower-dimensional space. Again the n-dimensional linear coefficients

$$\hat{\beta}_k^* = (\hat{\Sigma}^* + \lambda \mathbf{I})^{-1} \hat{\mu}_k^* \tag{25}$$

are mapped back to p-dimensions via $\hat{\beta}_k = \mathbf{Q}_1 \hat{\beta}_k^*$.

In this case further simplification is possible by diagonalizing $\hat{\Sigma}^*$ using the

[2]In fact, one can view RDA estimates as an instance of penalized optimal scoring [Hastie et al., 1995, 2001], for which there is an optimization problem of the form (5)

SVD. This allows one to efficiently compute the solutions for a series of values of λ without inverting matrices each time; see Guo et al. [2003] for more details.

RDA can also provide class probability estimates

$$\hat{\Pr}(y = k|x; \lambda) = \frac{e^{\delta_k(x;\lambda)}}{\sum_{j=1}^{K} e^{\delta_j(x;\lambda)}} \tag{26}$$

From (26) it is clear that the models used by RDA and multinomial regression (17) are of the same form; they both have linear discriminant functions, but the method for estimating these differ. This issue is taken up in Hastie et al. [2001, Chapter 4]. On these data RDA slightly outperformed multinomial regression (12 vs 15 test errors).

Regularized mixture discriminant analysis [Hastie and Tibshirani, 1996, Hastie et al., 2001] extend RDA in a flexible way, allowing several centers per class. The same computational tricks work here as well.

6 Discussion

There is one undesirable aspect to quadratically regularized linear models, for example, in the gene expression applications. The solutions $\hat{\beta}(\lambda)$ involve all the genes—no selection is done. An alternative is to use the so-called L_1 penalty $\lambda \sum_{j=1}^{p} |\beta_j|$ [Tibshirani, 1996], which causes many coefficients to be exactly zero. In fact, an L_1 penalty permits at most n nonzero coefficients [Efron et al., 2002, Zhu et al., 2003], which can be a problem if n is small. However, our computational trick to address the first issue only works with a quadratic penalty. Practice has shown that quadratically regularized models can still deliver good predictive performance. We have seen that SVMs are of this form, and they have become quite popular as classifiers. There have been several (ad hoc) approaches in the literature to select genes based on the size of their regularized coefficients (see Zhu and Hastie [2003] and references therein).

The models discussed here are not new; they have been in the statistics folklore for a long time, and many have already been used with expression arrays. The computational shortcuts possible with quadratically regularized linear models have also been discovered many times, often recently under the guise of "the kernel trick" in the kernel learning literature [Schölkopf and Smola, 2001]. Here we have shown that for linear models this device is totally transparent, and with a small amount of preprocessing all the models described here are computationally manageable with standard software.

References

D. Cox. Regression models and life tables (with discussion). *J. Royal. Statist. Soc. B.*, 74:187–220, 1972.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. Technical report, Stanford University, 2002.

J. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.

Y. Guo, T. Hastie, and R. Tibshirani. Regularized discriminant analysis and its application to micrarrays. Technical report, Statistics Department, Stanford University, 2003.

T. Hastie, A. Buja, and R. Tibshirani. Penalized discriminant analysis. *Annals of Statistics*, 23(1):73–102, 1995.

T. Hastie and R. Tibshirani. Discriminant analysis by gaussian mixtures. *J. Royal Statist. Soc. (Series B)*, 58:155–176, 1996.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning; Data mining, Inference and Prediction*. Springer Verlag, New York, 2001.

S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. Mesirov, T. Poggio, W. Gerald, M. Loda, E. Lander, and T. Golub. Multiclass cancer diagnosis using tumor gene expression signature. *PNAS*, 98:15149–15154, 2001.

S. Rosset, J. Zhu, and T. Hastie. Margin maximizing loss functions. In *Neural Information Processing Systems*, 2003. to appear.

B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. MIT Press, 2001.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc. B.*, 58:267–288, 1996.

R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Class prediction by nearest shrunken centroids, with applications to dna microarrays. *Statistical Science*, pages xxx–xxx, 2003.

V. Tusher, R. Tibshirani, and G. Chu. Significance analysis of microarrays applied to transcriptional responses to ionizing radiation. *Proc. Natl. Acad. Sci. USA.*, 98:5116–5121, 2001.

V. Vapnik. *The Nature of Statistical Learning*. Springer-Verlag, 1996.

J. Zhu and T. Hastie. Classification of gene microarrys by penalized logistic regression. Technical report, Statistic Department, Stanford University, 2003.

J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. L1 norm support vector machines. Technical report, Stanford University, 2003.