A Bayesian Approach to Variable Selection when the Number of Variables is Very Large

Harri T. Kiiveri

Abstract

In this paper, we present a rapid Bayesian variable selection technique which can be used when the number of variables is *much* greater than the number of samples. The method can handle tens of thousands of variables, such as might be measured using biological array technologies. A general formulation is first given, followed by specific details for the class of generalised linear models.

Keywords: Bayesian; Jeffreys hyperprior; posterior; variable selection; EM algorithm; generalised linear models; survival analysis

1 Introduction

Traditional methods of variable selection for statistical models include backward and forward stepwise procedures, and all subsets calculations using branch and bound algorithms, see for example [19]. Typically some criterion such as LAIC or BICE is used to guide the selection process. These stepwise methods have also been implemented in software packages such as R and Splus for more general models than linear regression, *e.g.* generalised linear models.

These traditional methods were implicitly designed for situations where the number of variables is less than the number of observations, and the number of variables was at most of the order of hundreds. Unfortunately, these methods do not cope well with large numbers of variables, say of the order of ten thousand, or when the number of observations is less than the number of variables. In these circumstance they either fail completely, or, even if they can be modified to work, require such a huge computational effort that they are impractical to use.

More recently, Bayesian variable selection methods based on Markov chain Monte Carlo methods have been developed, see for example [4, 13, 21, 22]. These have some attractive properties; however, aside from other issues, these methods are computationally intensive and do not scale up well to problems with ten thousand variables or more.

With the advent of microarray technologies, variable selection problems with ten thousand variables and hundreds of observations are becoming quite common, with the likelihood that the problem sizes will scale up at least one order of magnitude in the near future. Clearly, new methods are required to handle these large problems.

With this background in mind, we present here an automated method for eliminating redundant parameters from statistical models. The general method is presented first, followed by the special case when parameter elimination corresponds to variable selection in generalised linear models. This method can be applied when the number of parameters is much greater than the number of observations as well as in the usual case when the number of parameters is less than the number of observations.

In Section 2 we describe the general algorithm for the situation when there are two sets of parameters β and φ. In this case there is a prior expectation that many components of β are zero but not those of φ. For example, the β might be a large set of parameters such as might occur in a matrix factorisation and the φ might be a scale parameter or a shape parameter.

In Section 3 we consider an important special case of the algorithm, namely generalised linear models, in which a response, discrete or continuous, is explained by a set of covariates. In this case, eliminating (setting to zero) components of β corresponds to selecting relevant covariates or components and discarding the rest.

One application is to biological array data, where each biological array has a response associated with it, such as disease class or a continuous measurement of response to treatment. We seek to find (a small number of) components of the biological array data which explain or predict the response. Another application area is in spectroscopy, where spectra are measured over a large number of wavelengths and it is desired to predict sample properties of interest from the observed spectrum.

In the following, N denotes the number of samples, and vectors such as y, z and μ have components y_i, z_i and μ_i for $i = 1, \ldots, N$. Vector multiplication and division is defined component-wise and $\Delta(\cdot)$ denotes a diagonal matrix whose diagonals are equal to the argument. We also use $||\cdot||$ to denote Euclidean norm.

2 General algorithm for parameter selection

Consider a likelihood for some data y which is a function of a $p \times 1$ parameter vector β, many components of which are *a priori* expected to be zero, and a $q \times 1$ vector of parameters φ (not expected to be zero); note that q could be zero. We want a sparse model representation with as many components of β zero as possible.

The work of Figueiredo [10, 11] can be extended to handle this general problem. Basically, Figueiredo formulated a hierarchical prior for the regression parameters in the standard regression model as well as for the probit regression model for binary data. This prior had a Jeffreys hyperprior and strongly favoured regression parameters being zero. By using the trick of introducing a latent variable, he was able to construct an efficient EM algorithm for maximising the "posterior" distribution of the regression parameters. This posterior had discontinuous derivatives at any point where a component of beta was zero and would have caused problems in maximising the posterior directly. A natural by product of the maximisation was the elimination of redundant variables.

Following Figueiredo, we specify a prior for the parameters β by introducing a $p \times 1$

vector of hyperparameters ν^2. This prior is of the form

$$p(\beta) = \int_{\nu^2} p(\beta|\nu^2)\, p(\nu^2)\, d\nu^2, \tag{1}$$

where $p(\beta|\nu^2)$ is $N(0, \mathrm{diag}\{\nu^2\})$ and $p(\nu^2) \propto \prod_{i=1}^{p} 1/\nu_i^2$ is a Jeffreys prior for ν^2, [16]. We choose an uninformative prior for φ, although the following can be easily modified to include an informative prior. Writing $L(y|\beta\varphi)$ for the likelihood function, in this Bayesian framework the posterior distribution of β, φ and ν given y is

$$p(\beta, \varphi, \nu^2|y) \propto L(y|\beta\varphi)p(\beta|\nu^2)p(\nu^2). \tag{2}$$

By treating ν^2 as a vector of missing data, the EM algorithm [6] may be used to maximise (2) to produce maximum *a posteriori* estimates of β and φ. The prior above is such that the maximum *a posteriori* estimates will tend to be sparse; *i.e.* if a large number of parameters are redundant, many components of β will be zero. The algorithm is stated below.

2.1 EM algorithm for the general problem

To implement the EM Algorithm, we need to perform the so-called *E step* and *M step*. In the following, we start by initialising the algorithm, then perform the *E step*, which provides a function to maximise in the *M step*. Newton-Raphson iterations are used to carry out the *M step*, see [17]. After the *M step*, current values of φ are updated. Parameter values which fall below a threshold during the iterations are eliminated from the model, *i.e.* are fixed at zero.

1. Set $n = 0$, $S_0 = \{1, 2, \ldots, p\}$, initialise $\varphi^{(0)}$, β^* and put $\varepsilon = 10^{-5}$ (say)

2. Define

$$\beta_i^{(n)} = \begin{cases} \beta_i^*, & i \in S_n \\ 0 & \text{otherwise,} \end{cases}$$

and at iteration n, define P_n to be a matrix of zeroes and ones defined from the identity matrix of the same dimension as β by deleting columns corresponding to components of β which are zero. It is easy to see that

$$\begin{aligned} \gamma &= P_n^T \beta \qquad & \beta &= P_n \gamma \\ \gamma^{(n)} &= P_n^T \beta^{(n)} \qquad & \beta^{(n)} &= P_n \gamma^{(n)}, \end{aligned} \tag{3}$$

where the nonzero elements of $\beta^{(n)}$ are $\gamma^{(n)}$.

3. Perform the *E step* by calculating

$$\begin{aligned} Q(\beta|\beta^{(n)},\varphi^{(n)}) &= E\{\log p(\beta,\varphi,\nu|y)|y,\beta^{(n)},\varphi^{(n)}\} \\ &= L(y|\beta,\varphi^{(n)}) - 0.5(||\beta/\beta^{(n)}||^2), \end{aligned} \tag{4}$$

where L is the log likelihood function of y. The expectation is over ν^2. Using $\beta = P_n\gamma$ and $\beta^{(n)} = P_n\gamma^{(n)}$, Equation (4) can be written as

$$Q(\gamma|\gamma^{(n)},\varphi^{(n)}) = L(y|P_n\gamma,\varphi^{(n)}) - 0.5(||\gamma/\gamma^{(n)}||^2). \tag{5}$$

4. Perform the *M step*, which involves finding the maximum of (5) over γ. This can be done with Newton-Raphson iterations as follows. Set $\gamma_0 = \gamma^{(n)}$ and for $r = 0,1,2,\ldots,\gamma_{r+1} = \gamma_r + \alpha_r\delta_r$, where α_r is chosen by a line search algorithm to ensure that $Q(\gamma_{r+1}|\gamma^{(n)},\varphi^{(n)}) > Q(\gamma_r|\gamma^{(n)},\varphi^{(n)})$, and

$$\delta_r = \Delta(\gamma^{(n)})\left[-\Delta(\gamma^{(n)})\frac{\partial^2 L}{\partial^2\gamma_r}\Delta(\gamma^{(n)}) + I\right]^{-1}\left(\Delta(\gamma^{(n)})\frac{\partial L}{\partial\gamma_r} - \frac{\gamma_r}{\gamma^{(n)}}\right), \tag{6}$$

where $\partial L/\partial\gamma_r = P_n'\partial L/\partial\beta_r$, $\partial^2 L/\partial^2\gamma_r = P_n'\partial^2 L/\partial^2\beta_r P_n = P_n'\partial^2 L/\partial^2\beta_r P_n$. Equation (6) is simply the Newton-Raphson algorithm involving the first and second derivatives of (5) with respect to γ after some algebraic manipulation. Note the regularisation of the second derivative matrix induced by the prior.

5. Maximise (5) as a function of φ given the current estimate of β. Let γ^* be the value of γ_r when some convergence criterion is satisfied, *e.g.* $||\gamma_r - \gamma_{r+1}|| < \varepsilon$ (for example 10^{-5}). Define $\beta^* = P_n\gamma^*$, $S_{n+1} = \{i : |\beta_i^*| > \max_j(|\beta_j^*|\varepsilon_1)\}$ where ε_1 is a small constant, say 10^{-5}. The set S_{n+1} identifies variables which are still in the model. Now set $n = n+1$ and choose $\varphi^{(n+1)} = \varphi^{(n)} + \kappa_n(\varphi^* - \varphi^{(n)})$, where φ^* is a (local) maximum which satisfies $\partial/\partial\varphi L(y|P_n\gamma^*,\varphi) = 0$ and κ_n is a damping factor such that $0 < \kappa_n \leq 1$.

6. Check convergence. If $||\gamma^* - \gamma^{(n)}|| < \varepsilon_2$ where ε_2 is suitably small, then stop; otherwise, go to step 2 above.

For the general case, modifications are required if the regularised matrix in (6) is indefinite. The term $\partial^2 L/\partial^2\gamma_r$ in step 4 above can also be replaced by its expectation $E[\partial^2 L/\partial^2\gamma_r]$; we do this in Section 3 below.

2.2 Variable selection in generalised linear models

An important special case of the model and algorithm described above is generalised linear models (GLMs, see [20]). In the notation in the 1985 GLIM System Release manual, a GLM has likelihood function

$$L = \log p(y|\beta,\varphi) = \sum_{i=1}^{N}\left\{\frac{y_i\theta_i - b(\theta_i)}{a_i(\varphi)} + c(y_i,\varphi)\right\}, \tag{7}$$

where $y = (y_1,\ldots,y_n)^T$ and $a_i(\varphi) = \varphi/w_i$, with the w_i being a fixed set of known weights and φ a single scale parameter. We also have

$$E\{y_i\} = b'(\theta_i) \tag{8}$$

$$\mathrm{Var}\{y_i\} = b''(\theta_i)a(\varphi) = \tau_i^2 a_i(\varphi). \tag{9}$$

Each observation has a set of covariates x_i and a linear predictor $\eta_i = x_i^T\beta$. The relationship between the mean of the i^{th} observation μ_i and its linear predictor is given by the link function $\eta_i = g(\mu_i) = g(b'(\theta_i))$. The inverse of the link is denoted by h, *i.e.* $\mu_i = b'(\theta_i) = h(\eta_i)$. In summary, in addition to the scale parameter, a GLM can be specified by four components:

- the likelihood or (scaled) deviance function
- the link function
- the derivative of the link function
- the variance function.

Some common and well known examples of GLMs are given in table 1.

Table 1: Some examples of common GLMs

Distribution	Link function $g(\mu)$	Derivative of link function	Variance function	Scale parameter
Gaussian	μ	1	1	yes
Binomial	$\log(\mu/(1-\mu))$	$1/(\mu(1-\mu))$	$\mu(1-\mu)/n$	no
Poisson	$\log(\mu)$	$1/\mu$	μ	no
Gamma	$1/\mu$	$-1/\mu^2$	μ^2	yes
Inverse Gaussian	$1/\mu^2$	$-2/\mu^3$	μ^3	yes

For generalised linear models, it can be shown that

$$\frac{\partial L}{\partial \beta} = X^t\left\{\Delta\left(\frac{1}{\tau_i^2}\frac{\partial\mu_i}{\partial\eta_i}\right)\left(\frac{y_i-\mu_i}{a_i(\varphi)}\right)\right\}, \tag{10}$$

where X is the N by p matrix with i^{th} row x_i^T and

$$E\left\{\frac{\partial^2 L}{\partial\beta^2}\right\} = -E\left\{\frac{\partial L}{\partial\beta}\frac{\partial L}{\partial\beta}^T\right\}. \tag{11}$$

This can be written as

$$\frac{\partial L}{\partial\beta} = X^T V^{-1}(\frac{\partial\eta}{\partial\mu})(y-u) \tag{12}$$

$$E\left\{\frac{\partial^2 L}{\partial\beta^2}\right\} = -X^t V^{-1} X, \tag{13}$$

where $V = \Delta(a_i(\varphi)\tau_i^2(\partial\eta_i/\partial\mu_i)^2)$.

3 EM algorithm for variable selection in GLMs

A description of the EM algorithm follows for the special case of generalized linear models. The algorithm is of the same form as in Section 2, however we give more details regarding the choice of initial value and the calculation of first and second derivatives.

1. Set $n = 0, S_0 = \{1,2,\ldots,p\}$, $\varphi^{(0)}$, and $\varepsilon = 10^{-5}$ (say). If $p \leq N$ compute initial values of β^* by

$$\beta^* = (X^t X + \lambda I)^{-1} X^t g(y+\zeta); \tag{14}$$

if instead $p > N$, then compute initial values of β^* by

$$\beta^* = \frac{1}{\lambda}(I - X^t(X^t X + \lambda I)^{-1})X^t g(y+\xi), \tag{15}$$

where the ridge parameter λ satisfies $0 < \lambda \leq 1$ (say) and ζ is small and chosen so that the link function is well-defined at $y+\zeta$. Cross-validation [14] could be used to estimate λ.

2. Define

$$\beta_i^{(n)} = \begin{cases} \beta_i^*, & i \in S_n \\ 0 & \text{otherwise} \end{cases}$$

and let P_n be a matrix of zeroes and ones such that the nonzero elements $\gamma^{(n)}$ of $\beta^{(n)}$ satisfy

$$\gamma^{(n)} = P_n^T\beta^{(n)}, \qquad \beta^{(n)} = P_n\gamma^{(n)}$$
$$\gamma = P_n^T\beta, \qquad \beta = P_n\gamma.$$

3. Perform the *E step* by calculating

$$\begin{aligned} Q(\beta|\beta^{(n)},\varphi^{(n)}) &= E\{\log p(\beta,\varphi,\nu|y)|y,\beta^{(n)},\varphi^{(n)}\} \\ &= L(y|\beta,\varphi^{(n)}) - 0.5(||\beta/\beta^{(n)}||^2), \end{aligned} \tag{16}$$

where L is the GLM log likelihood function of y. Since $\beta = P_n\gamma$ and $\beta^{(n)} = P_n\gamma^{(n)}$, Equation (16) can be written as

$$Q(\gamma|\gamma^{(n)},\varphi^{(n)}) = L(y|P_n\gamma,\varphi^{(n)}) - 0.5(||\gamma/\gamma^{(n)}||^2) \tag{17}$$

4. Perform the *M step*. This can be done with Newton-Raphson iterations as follows. Set $\gamma_0 = \gamma^{(n)}$; for $r = 0,1,2,\ldots$, $\gamma_{r+1} = \gamma_r + \alpha_r\delta_r$, where α_r is chosen by a line search algorithm to ensure $Q(\gamma_{r+1}|\gamma^{(n)},\varphi^{(n)}) > Q(\gamma_r|\gamma^{(n)},\varphi^{(n)})$. For $p \leq N$, use

$$\delta_r = \Delta(\gamma^{(n)})[Y_n^T V^{-1} Y_n + I]^{-1}(Y_n^T V^{-1} z_r - \frac{\gamma_r}{\gamma^{(n)}}), \tag{18}$$

where

$$\begin{aligned} Y^T &= \Delta(\gamma^{(n)})P_n^T X^T \\ V &= \Delta\left(a_i(\varphi)\tau_i^2(\frac{\partial\eta_i}{\partial\mu_i})^2\right) \\ z &= \frac{\partial\eta}{\partial\mu}(y-\mu) \end{aligned}$$

and the subscript r denotes that these quantities are evaluated at $\mu_r = h(XP_n\gamma_r)$. For $p > N$, use

$$\delta_r = \Delta(\gamma^{(n)})\left[I - Y_n^T(Y_nY_n^T + V_r)^{-1}Y_n\right](Y_n^T V_r^{-1} z_r - \frac{\gamma_r}{\gamma^{(n)}}), \tag{19}$$

with V_r and z_r defined as before.

5. Let γ^* be the value of γ_r when some convergence criterion is satisfied, for example $||\gamma_r - \gamma_{r+1}|| < \varepsilon$ (*e.g.* 10^{-5}). Define $\beta^* = P_n\gamma^*$, $S_{n+1} = \{i : |\beta_i^*| > \max_j(|\beta_j^*|\varepsilon_1)$, where ε_1 is a small constant, say 10^{-5}. Set $n = n+1$ and choose $\varphi^{n+1} = \varphi^n + \kappa_n(\varphi^* - \varphi^n)$, where φ^* satisfies $\partial/\partial\varphi L(y|P_n\gamma^*,\varphi) = 0$ and κ_n is a damping factor such that $0 < \kappa_n \leq 1$. In some cases the scale parameter may be known, or this equation can be solved explicitly to get an updating equation for φ.

6. Check convergence. If $||\gamma^* - \gamma^{(n)}|| < \varepsilon_2$ for ε_2 suitably small, then stop; otherwise, go to step 2 above.

4 Remarks

1. The algorithm can be implemented to be $O(\min(N^3, p^3))$. Differentiation of (5) with respect to γ gives

$$\frac{\partial Q}{\partial \gamma} = \frac{\partial L}{\partial \gamma} - \frac{\gamma}{(\gamma^{(n)})^2}. \tag{20}$$

By the definition of the algorithm in Section 2, $\gamma^{(n+1)}$ is defined so that the left hand side of (20) is zero. Hence, if the sequence $(\gamma^{(n)}, \varphi^{(n)})$ converges, then from

$$(\gamma^{(n)})^2 \left(\frac{\partial L}{\partial \gamma^{(n+1)}} \right) = \gamma^{(n+1)}$$

we can see that redundant parameters which are still in the model but have yet to cross the threshold for omission approach zero at a quadratic rate. This observation is due to Dr. Frank De Hoog (personal communication) and is mirrored in the observed performance of the algorithm.

2. The selection of initial values is important, as values too close to zero can result in the solution $\beta = 0$. It also appears that multiple local maxima exist. The initial value is chosen so as to get a perfect fit to the training data if possible. The algorithm can then be viewed as sequentially throwing out variables which do not affect the fit, or cause the least degradation to the fit.

3. Integrating the prior in (2) over ν we obtain

$$p(\beta) \propto \prod_{j=1}^{p} |\beta_j|^{-1}.$$

Hence, if the likelihood evaluated at $\beta = 0$ is positive, the posterior will be improper. Use of Markov chain Monte Carlo (MCMC) to simulate from such a posterior requires caution, see for example [12].

4. Figueiredo [11] shows that replacing the Jeffreys prior in (1) by the prior

$$p(\nu_i^2|\gamma) = \exp(-\nu_i^2/\gamma)/\gamma$$

gives

$$p(\beta_i|\gamma) \propto \exp(-|\beta_i|\sqrt{2/\gamma}),$$

which is the prior used in the Lasso technique [23]. The algorithms described above have a simple modification to implement this model. Instead of using

$$E\{\nu_i^{-2}|\beta_i\} = \frac{1}{|\beta_i|^2}$$

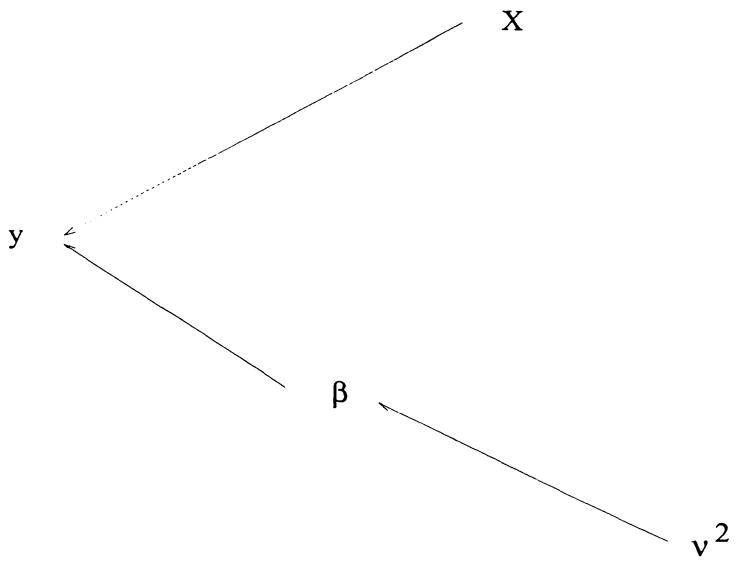

Figure 1: Graphical representation of factorisation of joint density in GLMs

in the *E step* at (4) and (16), use

$$E\{\nu_i^{-2}|\beta_i\} = \frac{1}{|\beta_i|}\sqrt{2/\gamma}.$$

The modifications to the Q functions in (4) and (15) should be clear. This requires the specification of a hyperparameter γ, something which is not required for the Jeffreys prior on ν. It is possible to give a general class of proper priors which includes as a special case the Lasso prior and as a limiting case the model (1) (in preparation).

5. The joint density for the GLMs can be represented graphically as in Figure 1. The *E step* in the EM algorithms described above does not involve y because of the conditional independence of ν^2 and y given β. This means that the algorithm can be applied for a wide variety of different likelihoods.

Another variation is to treat β as missing. With appropriate choice of hyperprior and likelihoods, this treatment gives algorithms for relevance vector machines, see [24]. However, approximations are usually required to do the *E step* since this now depends on y.

6. The algorithm in Section 3 can also be used for quasi-likelihood methods as described in [26] and [18].

7. The matrix X of covariates can be replaced by a matrix K with ij^{th} element k_{ij} and $k_{ij} = \kappa(x_i - x_j)$ for some kernel function κ. This matrix can also be augmented

with a vector of ones. Some possible kernels, including radial basis function kernels, are described in [9]. This treatment opens the possibility of fitting general smooth, as opposed to merely linear, functions of the covariates.

8. Our experience with the algorithm suggests that it is sometimes a little over-enthusiastic in throwing out variables. It is useful to keep a history of variables included in the model as iterations proceed, and to consider sets of variables one or two sets back from the final solution as well. The algorithm can also be used to perform an initial screening of variables for some other procedure by stopping iterations when some subset size is approached *e.g.* 50 variables or when the initial "perfect" fit degrades significantly.

9. By projecting variables not chosen onto the space spanned by a set of chosen variables and then clustering, equivalence classes of important variables can be identified. Alternative solutions can be explored by using a sequence of runs in which the variables chosen in the previous run and those equivalent to them are omitted from consideration in the next run.

5 Examples

In this section, we present examples of the use of these algorithms for some common GLMs and for survival analysis. In each case, we use the version of the algorithm in Section 3.1 with Jeffreys hyperprior (no hyperparameters required). Execution time was typically less than one minute when run in R on a computer with a Pentium III 500 MHz processor and 256 Mb of RAM.

5.1 Standard linear regression model

The algorithm for linear regression is described in [11]. We include it here as an example of a generalised linear model. Consider the sugars data analysed in [3]. The data consist of 125 training observations, where each observation consists of a (transformed) spectrum measured at 700 wavelengths. There is a validation set of 21 observations. The "responses" to be predicted are the percentage composition of three sugars, sucrose, glucose and fructose, in water. We analyse each sugar separately for illustrative purposes here.

The standard regression model is well-known to be a generalised linear model with

- Link function: $g(\mu) = \mu$
- Derivative of link function: $\frac{\partial \eta}{\partial \mu} = 1$
- Variance function: $\tau^2 = 1$
- Scale parameter $\varphi = \sigma^2$
- Deviance (likelihood function): $-\frac{N}{2}\log(\sigma^2) - 0.5\sum_{i=1}^{N}(y_i - \mu_i)^2$

- Updating formula for σ^2 given by

$$(\sigma^2)^{(n+1)} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \mu_i^*)^2,$$

where μ_i^* is the mean evaluated at β^* in step 5 of the algorithm.

For the linear regression model, we substitute the deviance function defined above for L in (16). Using the above, we can evaluate the terms in (18) as

$$\begin{aligned} Y_n^T &= \Delta(\gamma^{(n)})P_n^T X^T \\ V &= \sigma_n^2 I \\ z_r &= (y - \mu_r) \end{aligned}$$

$$\delta_r = \Delta(\gamma^{(n)})[Y_n^T Y_n + \sigma_n^2 I]^{-1}\left(Y_n^T(y-\mu_r) - \sigma_n^2\frac{\gamma_r}{\gamma^{(n)}}\right). \tag{21}$$

The iterations (21) are basically ridge regressions. An expression for the case when p is greater than N, which involves inversion of a smaller matrix, can be obtained from (19).

For sucrose and glucose, the algorithm in Section 3.1 selected 9 variables (wavelengths), including a constant term. For fructose, the algorithm selected 5 wavelengths with no constant term. The chosen wavelengths in nanometres are given below.

Sucrose 1896 1904 1908 1960 1968 2248 2250 2284

Glucose 1882 1908 1950 1958 1968 2008 2280 2332

Fructose 1908 2082 2254 2256 2330

Results for mean square error (MSE) are given in Table 2.

Table 2: Results on training and validation data

Sugar	Training MSE	Validation MSE
Sucrose	0.10	2.34
Glucose	0.09	0.36
Fructose	0.13	0.38

The mean square error for sucrose on the validation set is much larger than that of the other two sugars. A look at the data suggests that there is a bias in the validation set in the water absorption region of the spectrum as compared to the training data. Deleting

the corresponding wavelengths (1748 to 2498 inclusive) and re-running the algorithm for sucrose reduced the mean square error on the validation set to 1.11 and produced a model with 5 wavelengths, namely 1406, 1756, 1772, 1792, and 2316. Although the validation mean square errors are somewhat larger than those reported in [3], the predictions are quite good and make use of smaller sets of wavelengths than those chosen by the selection method in [3].

5.2 Logistic regression example

We illustrate logistic regression with the data set of [2]. There are $p = 4026$ genes and $N = 36$ samples. In the following, DLBCL refers to *diffuse large B-cell lymphoma*. The samples have been classified into two disease types: GC B-like DLBCL (21 samples) and Activated B-like DLBCL (15 samples). We use this set to illustrate how the above methodology may be used for rapidly identifying genes which are potentially diagnostic of different disease types. The data have been used to define the classes, see [2]; however, we simply use the data set to illustrate the method here.

Logistic regression is a generalised linear model with response y here being the disease class labelled 0 or 1. We also have

- Link function: $g(\mu) = \log(\mu/(1-\mu))$
- Derivative of link function: $1/(\mu(1-\mu))$
- Variance function: $\mu(1-\mu)$
- Scale parameter $\varphi = 1$
- Deviance (likelihood function): $\sum_{i=1}^{N}\{y_i \log(\mu_i) + (1-y_i)\log(1-\mu_i)\}$
- No updating formula is required for the scale parameter.

For the logistic regression model, we substitute the Deviance function defined above for L in (16). Using the above, we can evaluate the terms in (18) as

$$\begin{aligned} Y_n^T &= \Delta(\gamma^{(n)})P_n^T X^T \\ V &= \Delta(\mu_r^{-1}(1-\mu_r)^{-1}) \\ z_r &= \mu_r^{-1}(1-\mu_r)^{-1}(y-\mu_r) \end{aligned}$$

and

$$\delta_r = \Delta(\gamma^{(n)})[Y_n^T \Delta(\mu_r(1-\mu_r)))Y_n + I]^{-1}(Y_n^T(y-\mu_r) - \frac{\gamma_r}{\gamma^{(n)}}). \tag{22}$$

The iterations (22) are once again basically ridge regressions. The algorithm identified 3 relevant genes. The classification accuracy on the training data is given below. This is a much smaller set of genes than the set used by Alizadeh *et al.* [2] to construct the classes.

Table 3: Classification accuracy for the 3 gene logistic regression model

	Predicted class 1	Predicted class 2
True class 1	20	1
True class 2	2	13

5.3 Another logistic regression example

The dataset for this example [15] is available from `http://www-genome.wi.mit.edu/MPR/data_set_ALL_AML.html`. The training data consist of 38 observations with 7129 variables (genes). The validation set contains 34 observations. The response variable is the leukemia class: acute myeloid leukemia (AML) or acute lymphoblastic leukemia (ALL). The data set available over the web includes more genes than those used in the original analysis of [15]. The dataset contains some controls; however, to test the algorithm we include all the data in the data set.

The algorithm retains 4 genes out of the 7129 considered. These are

- 1763 Thymosin beta-4 mRNA
- 1779 MPO Myeloperoxidase
- 2402 Azurocidin gene
- 6201 Interleukin-8 precursor.

This set of genes gives perfect separation of the classes in the training data. An analysis of equivalent sets suggests that gene 6201 can be interchanged with gene 6200, namely Interleukin 8 (IL8) gene. These genes are biologically meaningful in this context.

Table 4 shows results for the selected model applied to the validation set.

Table 4: Validation accuracy for the logistic regression model with 4 selected genes

	Predicted ALL	Predicted AML
True ALL	20	0
True AML	3	11

We also performed an analysis similar to [7] whereby the data was randomly divided into training and test sets in the ratio 2:1. For comparison purposes, we used the 3157 genes used in [7]. The variable selection was run for each training set, and predictions were made for the corresponding test set in a total of 150 runs. All 3157 genes were considered in each run, there was no preselection of genes. The median number of misclassifications observed was 2 with a maximum of 5 and minimum of 0. When

we used a more general prior somewhat between the lasso and (1), the median number of misclassifications was 1.5 with maximum 3 and minimum 0. The mean number of variables chosen was 3. For details see [8].

5.4 Poisson regression example

We use the data set in Section 5.2 to also illustrate gene selection in Poisson regression.

We artificially created a new gene (gene number 1) and a Poisson response for each array with mean given by the expression value of gene 1. This new gene was added to the previous data matrix. Hence, there are 4027 "genes" and 36 samples in this case. The response has a Poisson distribution.

Poisson regression is a generalised linear model with

- Link function: $g(\mu) = log(\mu)$
- Derivative of link function: $1/\mu$
- Variance function: $\tau^2 = \mu$
- Scale parameter $\varphi = 1$
- Deviance (likelihood function): $\sum_{i=1}^{N} \{y_i \log(\mu_i) - \mu_i\}$
- No updating formula is required for the scale parameter.

The algorithm required 5 iterations to correctly identify "gene" 1 as the relevant gene.

5.5 Cox proportional hazards model

We apply a version of the general algorithm in Section 2 to the survival data of Alizadeh *et al.* [2] (available at `http://llmpp.nih.gov/lymphoma/data.shtml`). A parametric version of this can also be fitted as a GLM using a Poisson model, see [1]. In this application, two observations (patients DLCL-0051 and DLCL-0052) are omitted because there is no survival information available for them. The data consist of cDNA microarray measurements on 4026 genes from 40 patients, survival times for each patient and a censoring indicator.

A Cox proportional hazards model [5] is fitted with an initial 4026 explanatory variables (*i.e.* genes) that are rapidly whittled down by the algorithm to just three explanatory genes. The explanatory genes identified by the algorithm are GENE3797X, GENE3302X and GENE356X. These are

- Immunoglobulin heavy chain V(H)5 pseudogene L2-9 transcript
- adenosine deaminase – this is a target for some drugs used to treat lymphoma
- AIM2 – involved with interferon induction and cell fate.

The selected genes are biologically meaningful. More details about this analysis and a simple prognostic indicator can be found in [25].

6 Conclusion

The algorithms described above seem promising in situations where there is little prior knowledge concerning the relationship of a large number of variables to a response of interest. They are fast and can be scaled up to handle *large* numbers of variables. They can also provide a useful screening tool to weed out apparently unimportant parameters or variables prior to an analysis by some other method.

A concern in this context is the production of results which are purely artifacts due to the large number of variables to choose from. Another concern is the influence of individual high dimensional observations when the number of samples is relatively small. As regards the former, permutation tests and the use of validation data sets have confirmed that the results so far are unlikely to be artifacts. In limited testing to date with biological arrays, the algorithms have produced biologically meaningful and apparently new results. A key feature is the consistent identification of smaller sets of variables with performance similar to the larger sets reported by other analyses. A similar statement can be made for spectroscopic data. Concerning the stability of the models selected, leave one out cross-validation calculations have so far demonstrated a high degree of stability in the chosen models. However, more work is required to test these ideas.

We are currently exploring other applications, such as logistic multi-class classification models.

The algorithms and analysis methods described here are protected by patents which are owned by CSIRO.

Acknowledgments

I would like to thank Dr. Frank De Hoog and Professor Phillip Brown for helpful insights and discussions.

Harri T. Kiiveri, CSIRO Mathematical and Information Sciences, The Leeuwin Centre, Floreat, Western Australia, harri.kiiveri@csiro.au

References

[1] M. Aitkin and C. Clayton. The fitting of exponential, Weibull and extreme value distributions to complex censored survival data using GLIM. *Applied Statistics*, 29:156–163, 1980.

[2] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson Jr, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, 2000.

[3] P. J. Brown. Measurement, regression and calibration. *Oxford University Press*, 1993.

[4] P. J. Brown, M. Vanucci, and T. Fearn. Bayes model averaging with selection of regressors. *Journal of the Royal Statistical Society B*, 64:519–536, 2002.

[5] D. R. Cox and D. Oakes. Analysis of survival data. *Chapman and Hall, London*, 1984.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–21, 1977.

[7] S. Dudoit, J. Fridyland, and T. P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97:77–87, 2002.

[8] R. Dunne. Classification of genes and arrays for microarray data. Technical report, Internal CMIS report, 2001.

[9] T. Evgeniou, M. Pontil, and T. Poggio. A unified framework for regularization networks and support vector machines. *MIT AI memo 1654*, 1999.

[10] M. Figueiredo. Adaptive sparseness using Jeffreys prior. *Neural Information Processing Systems - NIPS ' 2001, Vancouver, December 2001*, 2001.

[11] M. Figueiredo. Unsupervised sparse regression. In *MSRI Workshop on Nonlinear Estimation and Classification, Berkeley, CA*, March 2001. to appear.

[12] I. E. Gelfand and S. K. Sahu. Identifiability, improper priors, and Gibbs sampling for generalised linear models. *Journal of the American Statistical Association*, 94:247–253, 1999.

[13] E. I. George and R. E. McCulloch. Stochastic search variable selection. In *Markov Chain Monte Carlo in Practice*. Chapman and Hall, New York, 1996.

[14] G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215–223, 1979.

[15] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[16] S. Kotz and N. L. Johnson. Encyclopedia of statistical sciences. *Wiley, New York*, 4:639, 1983.

[17] D. G. Luenberger. Introduction to linear and nonlinear programming. *Addison-Wesley, Reading, Massachusetts*, 1973.

[18] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Chapman and Hall, second edition, 1989.

[19] A. J. Miller. *Subset selection in Regression*. Chapman and Hall, New York, 1990.

[20] J. A. Nelder and R. W. M. Weddeburn. Generalised linear models. *Journal of the Royal Statistical Society A*, 135:370–384, 1972.

[21] D. B. Phillips and A. F. M. Smith. Bayesian model comparison via jump diffusions. In *Markov Chain Monte Carlo in Practice*. Chapman and Hall, New York, 1996.

[22] N. Sha, M. Vanucci, and P. J. Brown. Bayesian variable selection in multinomial probit models with application to spectral data and DNA microarrays. *Submitted for publication*, 2002.

[23] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58:267–288, 1996.

[24] M. Tipping. The relevance vector machine. In *Advances in Neural Information Processing Systems*, volume 12, pages 652–658, 2000.

[25] A. C. Trajstman and H. T. Kiiveri. Applications of a rapid variable selection technique to microarray output and survival data generated from a study of B-cell lymphoma: gene discovery and survival prognosis. *CSIRO CMIS internal report*, 2002.

[26] R. W. M. Wedderburn. Quasi-likelihood functions, generalised linear models, and the Gauss-Newton method. *Biometrika*, 64:439–447, 1974.