J. R. Statist. Soc. B (2007)
69, *Part* 4, *pp.* 659–677

L_1-regularization path algorithm for generalized linear models

Mee Young Park

Google Inc., Mountain View, USA

and Trevor Hastie

Stanford University, USA

[Received February 2006. Final revision March 2007]

Summary. We introduce a path following algorithm for L_1-regularized generalized linear models. The L_1-regularization procedure is useful especially because it, in effect, selects variables according to the amount of penalization on the L_1-norm of the coefficients, in a manner that is less greedy than forward selection–backward deletion. The generalized linear model path algorithm efficiently computes solutions along the entire regularization path by using the predictor–corrector method of convex optimization. Selecting the step length of the regularization parameter is critical in controlling the overall accuracy of the paths; we suggest intuitive and flexible strategies for choosing appropriate values. We demonstrate the implementation with several simulated and real data sets.

Keywords: Generalized linear model; Lasso; Path algorithm; Predictor–corrector method; Regularization; Variable selection

1. Introduction

In this paper we propose a path following algorithm for L_1-regularized generalized linear models (GLMs). GLMs model a random variable Y that follows a distribution in the exponential family by using a linear combination of the predictors $\mathbf{x}'\beta$, where $\mathbf{x}$ and β denote vectors of the predictors and the coefficients respectively. The random and the systematic components may be linked through a non-linear function; therefore, we estimate the coefficient β by solving a set of non-linear equations that satisfy the maximum likelihood criterion

$$\hat{\beta} = \arg\max_{\beta}\{L(\mathbf{y};\beta)\}, \tag{1}$$

where L denotes the likelihood function with respect to the given data $\{(\mathbf{x}_i, y_i) : i = 1, \ldots, n\}$.

When the number of predictors p exceeds the number of observations n, or when insignificant predictors are present, we can impose a penalization on the L_1-norm of the coefficients for an automatic variable selection effect. Analogously to the lasso (Tibshirani, 1996) that added a penalty term to the squared error loss criterion, we modify criterion (1) with a regularization:

$$\hat{\beta}(\lambda) = \arg\min_{\beta}[-\log\{L(\mathbf{y};\beta)\} + \lambda\|\beta\|_1], \tag{2}$$

where $\lambda > 0$ is the regularization parameter. Logistic regression with L_1-penalization has been

Address for correspondence: Mee Young Park, Google Inc., 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA.
E-mail: meeyoung@google.com

 1369–7412/07/69659

introduced in Lokhorst (1999) and explored by several researchers, e.g. in Shevade and Keerthi (2003) and Genkin *et al.* (2004).

We introduce an algorithm that implements the predictor–corrector method to determine the entire path of the coefficient estimates as λ varies, i.e. to find $\{\hat{\beta}(\lambda) : 0 < \lambda < \infty\}$. Starting from $\lambda = \lambda_{\max}$, where $\lambda_{\max}$ is the largest λ that makes $\hat{\beta}(\lambda)$ non-zero, our algorithm computes a series of solution sets, each time estimating the coefficients with a smaller λ based on the previous estimate. Each round of optimization consists of three steps: determining the step size in λ, predicting the corresponding change in the coefficients and correcting the error in the previous prediction.

A traditional approach to variable selection is the forward selection–backward deletion method that adds or deletes variables in a greedy manner. L_1-regularization as in criterion (2) can be viewed as a smoother and 'more democratic' version of forward stepwise selection. By generating the regularization path rather than computing solutions at several fixed values of λ, we identify the order in which the variables enter or leave the model. Thus, we can find a regularized fit with any given number of parameters, as with the series of models from the forward stepwise procedure.

Efron *et al.* (2004) suggested an efficient algorithm to determine the exact piecewise linear coefficient paths for the lasso; see Osborne *et al.* (2000) for a closely related approach. The algorithm called *LARS* is also used for forward stagewise and least angle regression paths with slight modifications. Another example of a path following procedure is the support vector machine path; see Hastie *et al.* (2004). They presented a method of drawing the entire regularization path for the support vector machine simultaneously.

Unlike LARS or support vector machine paths, the GLM paths are not piecewise linear. We must select particular values of λ at which the coefficients are computed exactly; the granularity controls the overall accuracy of the paths. When the coefficients are computed on a fine grid of values for λ, the non-linearity of the paths is more visible. We propose a way to compute the exact coefficients at the values of λ at which the set of non-zero coefficients changes. This strategy yields a more accurate path in an efficient way than alternative methods and provides the exact order of the active set changes, which is important information in many applications, such as gene selection.

Rosset (2004) suggested a general path following algorithm that can be applied to any loss and penalty function with reasonable bounds on the domains and the derivatives. This algorithm computes the coefficient paths in two steps: changing λ and updating the coefficient estimates through a Newton iteration. Zhao and Yu (2004) proposed the *boosted lasso* that approximates the L_1-regularization path with respect to any convex loss function by allowing backward steps to forward stagewise fitting; whenever a step in forward stagewise fitting deviated from that of the lasso, the boosted lasso would correct the step with a backward move. When this strategy is used with the *negative log-likelihood* (of a distribution in the exponential family) loss function, it will approximate the L_1-regularized GLM path. As discussed by Zhao and Yu (2004), the step sizes along the path are distributed such that their method finds the exact solutions at uniformly spaced values of $\|\beta\|_1$, whereas Rosset's method computes solutions at uniformly spaced λ. Our method is more flexible and efficient than these two approaches; we estimate the largest λ that will change the current active set of variables and solve for the new set of solutions at the estimated λ. Hence, the step lengths are not uniform for any single parameter but depend on the data; at the same time, we ensure that the solutions are exact at the locations where the active set changes. We demonstrate the accuracy and the efficiency of our strategy in Section 3.2.

In the following sections, we describe and support our approach in more detail with examples and justifications. We present the details of the GLM path algorithm in Section 2. In Section 3,

our methods are illustrated with simulated and real data sets, including a microarray data set consisting of over 7000 genes. We illustrate an extension of our path following method to the Cox proportional hazards model in Section 4. We conclude with a summary and other possible extensions of our research in Section 5. Proofs for all the lemmas and theorems are provided in Appendix A.

2. Generalized linear model path algorithm

In this section, we describe the details of the GLM path algorithm. We compute the exact solution coefficients at particular values λ and connect the coefficients in a piecewise linear manner for solutions corresponding to other values of λ.

2.1. Problem set-up

Let $\{(\mathbf{x}_i, y_i) : \mathbf{x}_i \in \mathcal{R}^p, y_i \in \mathcal{R}, i = 1, \ldots, n\}$ be n pairs of p predictors and a response. Y follows a distribution in the exponential family with mean $\mu = E(Y)$ and variance $V = \mathrm{var}(Y)$. Depending on its distribution, the domain of y_i could be a subset of $\mathcal{R}$. GLMs model the random component Y by equating its mean μ with the systematic component η through a link function g:

$$\eta = g(\mu) = \beta_0 + \mathbf{x}'\beta. \tag{3}$$

The density function of Y is expressed as (McCullagh and Nelder, 1989)

$$L(y; \theta, \phi) = \exp\left\{ \frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi) \right\}. \tag{4}$$

$a(\cdot), b(\cdot)$ and $c(\cdot)$ are functions that vary according to the distributions. Assuming that the dispersion parameter ϕ is known, we are interested in finding the maximum likelihood solution for the natural parameter θ, and thus $(\beta_0, \beta')'$, with a penalization on the size of the L_1-norm of the coefficients ($\|\beta\|_1$). Therefore, our criterion with a fixed λ is reduced to finding $\boldsymbol{\beta} = (\beta_0, \beta')'$, which minimizes

$$l(\boldsymbol{\beta}, \lambda) = -\sum_{i=1}^{n} [y_i \, \theta(\boldsymbol{\beta})_i - b\{\theta(\boldsymbol{\beta})_i\}] + \lambda \|\beta\|_1. \tag{5}$$

Assuming that none of the components of β is 0 and differentiating $l(\boldsymbol{\beta}, \lambda)$ with respect to $\boldsymbol{\beta}$, we define a function H:

$$H(\boldsymbol{\beta}, \lambda) = \frac{\partial l}{\partial \boldsymbol{\beta}} = -\mathbf{X}'\mathbf{W}(\mathbf{y} - \boldsymbol{\mu}) \frac{\partial \eta}{\partial \mu} + \lambda \, \mathrm{sgn}\begin{pmatrix} 0 \\ \beta \end{pmatrix}, \tag{6}$$

where $\mathbf{X}$ is an $n \times (p+1)$ matrix including a column of 1s, $\mathbf{W}$ is a diagonal matrix with n diagonal elements $V_i^{-1}(\partial\mu/\partial\eta)_i^2$ and $(\mathbf{y} - \boldsymbol{\mu})\partial\eta/\partial\mu$ is a vector with n elements $(y_i - \mu_i)(\partial\eta/\partial\mu)_i$. Although we have assumed that none of the elements of β is 0, the set of non-zero components of $\boldsymbol{\beta}$ changes with λ, and $H(\boldsymbol{\beta}, \lambda)$ must be redefined accordingly.

Our goal is to compute the entire solution path for the coefficients $\boldsymbol{\beta}$, with λ varying from $\lambda_{\max}$ to 0. We achieve this by drawing the curve $H(\boldsymbol{\beta}, \lambda) = 0$ in $(p+2)$-dimensional space ($\boldsymbol{\beta} \in \mathcal{R}^{p+1}$ and $\lambda \in \mathcal{R}_+$). Rosset *et al.* (2004) provided sufficient conditions for the existence of a unique solution $\boldsymbol{\beta}(\lambda)$ that minimizes the convex function $l(\boldsymbol{\beta}, \lambda)$ for each $\lambda \in \mathcal{R}_+$. We first restrict our attention to the cases where the conditions are satisfied and present the algorithm; we suggest a strategy to extend the algorithm to the cases where the conditions do not hold in Section 2.4. In the former situation, a unique continuous and differentiable function $\boldsymbol{\beta}(\lambda)$, such that $H\{\boldsymbol{\beta}(\lambda), \lambda\} = 0$ exists

within each open range of λ that yields a certain active set of variables; the existence of such mappings ($\lambda \to \boldsymbol{\beta}(\lambda)$) can be shown by using the implicit function theorem (Munkres, 1991). We find the mapping $\boldsymbol{\beta}(\lambda)$ sequentially with decreasing λ.

2.2. Predictor–corrector algorithm

The predictor–corrector algorithm is one of the fundamental strategies for implementing numerical continuation (which has been introduced and applied in, for example, Allgower and Georg (1990) and Garcia and Zangwill (1981)). Numerical continuation has long been used in mathematics to identify the set of solutions to non-linear equations that are traced through a one-dimensional parameter. Among many approaches, the predictor–corrector method explicitly finds a series of solutions by using the initial conditions (solutions at one extreme value of the parameter) and continuing to find the adjacent solutions on the basis of the current solutions. We elaborate on how the predictor–corrector method is used to trace the curve $H(\boldsymbol{\beta}, \lambda) = 0$ through λ in our problem setting.

The following lemma provides the initialization of the coefficient paths.

Lemma 1. When λ exceeds a certain threshold, the intercept is the only non-zero coefficient: $\hat{\beta}_0 = g(\bar{y})$ and

$$H\{(\hat{\beta}_0, 0, \ldots, 0)', \lambda\} = 0 \qquad \text{for } \lambda > \max_{j \in \{1,\ldots,p\}} |\mathbf{x}_j' \hat{\mathbf{W}}(\mathbf{y} - \bar{y}\mathbf{1})\, g'(\bar{y})|. \tag{7}$$

We denote this threshold of λ as $\lambda_{\max}$. As λ is decreased further, other variables join the active set, beginning with the variable $j_0 = \arg\max_j |\mathbf{x}_j'(\mathbf{y} - \bar{y}\mathbf{1})|$. Reducing λ, we alternate between a predictor and a corrector step; the steps of the kth iteration are as follows.

Step 1: step length—determine the decrement in λ. Given λ_k, we approximate the next largest λ, at which the active set changes, namely λ_{k+1}.
Step 2: predictor step—linearly approximate the corresponding change in $\boldsymbol{\beta}$ with the decrease in λ; call it $\hat{\boldsymbol{\beta}}^{k+}$.
Step 3: corrector step—find the exact solution of $\boldsymbol{\beta}$ that pairs with λ_{k+1} (i.e. $\boldsymbol{\beta}(\lambda_{k+1})$), using $\hat{\boldsymbol{\beta}}^{k+}$ as the starting value; call it $\hat{\boldsymbol{\beta}}^{k+1}$.
Step 4: active set—test to see whether the current active set must be modified; if so, repeat the corrector step with the updated active set.

2.2.1. Predictor step

In the kth predictor step, $\boldsymbol{\beta}(\lambda_{k+1})$ is approximated by

$$\hat{\boldsymbol{\beta}}^{k+} = \hat{\boldsymbol{\beta}}^k + (\lambda_{k+1} - \lambda_k)\frac{\partial \boldsymbol{\beta}}{\partial \lambda} \tag{8}$$

$$= \hat{\boldsymbol{\beta}}^k - (\lambda_{k+1} - \lambda_k)(\mathbf{X}_A' \mathbf{W}_k \mathbf{X}_A)^{-1} \operatorname{sgn}\begin{pmatrix} 0 \\ \hat{\beta}^k \end{pmatrix}. \tag{9}$$

$\mathbf{W}_k$ and $\mathbf{X}_A$ denote the current weight matrix and the columns of $\mathbf{X}$ for the factors in the current active set respectively. $\boldsymbol{\beta}$ in the above equations are composed only of current non-zero coefficients. This linearization is equivalent to making a quadratic approximation of the log-likelihood and extending the current solution $\hat{\boldsymbol{\beta}}^k$ by taking a weighted lasso step (as in the LARS algorithm).

Define $f(\lambda) = H\{\boldsymbol{\beta}(\lambda), \lambda\}$; in the domain that yields the current active set, $f(\lambda)$ is 0 for all λ. By differentiating f with respect to λ, we obtain

$$f'(\lambda) = \frac{\partial H}{\partial \lambda} + \frac{\partial H}{\partial \boldsymbol{\beta}} \frac{\partial \boldsymbol{\beta}}{\partial \lambda} = 0, \tag{10}$$

from which we compute $\partial \boldsymbol{\beta}/\partial \lambda$.

The following theorem shows that the predictor step approximation can be arbitrarily close to the real solution by making $\lambda_k - \lambda_{k+1}$ small.

Theorem 1. Denote $h_k = \lambda_k - \lambda_{k+1}$, and assume that h_k is sufficiently small that the active sets at $\lambda = \lambda_k$ and $\lambda = \lambda_{k+1}$ are the same. Then the approximated solution $\hat{\boldsymbol{\beta}}^{k+}$ differs from the real solution $\hat{\boldsymbol{\beta}}^{k+1}$ by $O(h_k^2)$.

2.2.2. *Corrector step*

In the following corrector step, we use $\hat{\boldsymbol{\beta}}^{k+}$ as the initial value to find the $\boldsymbol{\beta}$ that minimizes $l(\boldsymbol{\beta}, \lambda_{k+1})$, as defined in equation (5) (i.e. that solves $H(\boldsymbol{\beta}, \lambda_{k+1}) = 0$ for $\boldsymbol{\beta}$). Any (convex) optimization method that applies to the minimization of a differentiable objective function with linear constraints may be implemented. The previous predictor step has provided a starting-point for the optimization, because $\hat{\boldsymbol{\beta}}^{k+}$ is usually close to the exact solution $\hat{\boldsymbol{\beta}}^{k+1}$, the cost of solving for the exact solution is low. The corrector steps not only find the exact solutions at a given λ but also yield the directions of $\boldsymbol{\beta}$ for the subsequent predictor steps.

We connect $\hat{\boldsymbol{\beta}}^{k+1}$ with $\hat{\boldsymbol{\beta}}^{k}$, forming the kth linear segment of the path. We justify this approach by showing that, if $\lambda_k - \lambda_{k+1}$ is small, then any point along the linear segment is close to the true path in some sense.

Theorem 2. If the solutions at λ_k and $\lambda_{k+1} = \lambda_k - h_k$, namely $\hat{\boldsymbol{\beta}}^{k}$ and $\hat{\boldsymbol{\beta}}^{k+1}$, are connected such that our estimate at $\lambda = \lambda_k - \alpha h_k$ for some $\alpha \in [0, 1]$ is

$$\hat{\boldsymbol{\beta}}(\lambda - \alpha h_k) = \hat{\boldsymbol{\beta}}^{k} + \alpha(\hat{\boldsymbol{\beta}}^{k+1} - \hat{\boldsymbol{\beta}}^{k}), \tag{11}$$

then $\hat{\boldsymbol{\beta}}(\lambda - \alpha h_k)$ differs from the real solution $\boldsymbol{\beta}(\lambda - \alpha h_k)$ by $O(h_k^2)$.

2.2.3. *Active set*

The active set $\mathcal{A}$ begins from the intercept as in lemma 1; after each corrector step, we check to see whether $\mathcal{A}$ should have been augmented. The following procedure for checking was justified and used by Rosset and Zhu (2003) and Rosset (2004):

$$\left| \mathbf{x}_j' \mathbf{W} (\mathbf{y} - \boldsymbol{\mu}) \frac{\partial \eta}{\partial \mu} \right| > \lambda \qquad \text{for any } j \in \mathcal{A}^c \Rightarrow \mathcal{A} \leftarrow \mathcal{A} \cup \{j\}. \tag{12}$$

We repeat the corrector step with the modified active set until the active set is not augmented further. We then remove the variables with zero coefficients from the active set, i.e.

$$|\hat{\beta}_j| = 0 \qquad \text{for any } j \in \mathcal{A} \Rightarrow \mathcal{A} \leftarrow \mathcal{A} \setminus \{j\}. \tag{13}$$

2.2.4. *Step length*

Two natural choices for the step length $\Delta_k = \lambda_k - \lambda_{k+1}$ are

(a) $\Delta_k = \Delta$, fixed for every k, or
(b) a fixed change L in L_1 arc length, achieved by setting $\Delta_k = L / \|\partial \boldsymbol{\beta}/\partial \lambda\|_1$.

As we decrease the step size, the exact solutions are computed on a finer grid of λ-values, and the coefficient path becomes more accurate.

We propose a more efficient and useful strategy:

(c) select the smallest Δ_k that will change the active set of variables.

We give an intuitive explanation of how we achieve this, by drawing on analogies with the LARS algorithm (Efron *et al.*, 2004). At the end of the kth iteration, the corrector step can be characterized as finding a weighted lasso solution that satisfies

$$-\mathbf{X}_A' \mathbf{W}_k (\mathbf{y} - \boldsymbol{\mu}) \frac{\partial \eta}{\partial \mu} + \lambda_k \operatorname{sgn}\begin{pmatrix} 0 \\ \beta \end{pmatrix} = 0.$$

This weighted lasso also produces the direction for the next predictor step. If the weights $\mathbf{W}_k$ were fixed, the weighted LARS algorithm would be able to compute the exact step length to the next active set changepoint. We use this step length, even though in practice the weights change as the path progresses.

Lemma 2. Let $\hat{\boldsymbol{\mu}}$ be the estimates of $\mathbf{y}$ from a corrector step, and denote the corresponding weighted correlations as

$$\hat{\mathbf{c}} = \mathbf{X}' \hat{\mathbf{W}} (\mathbf{y} - \hat{\boldsymbol{\mu}}) \frac{\partial \eta}{\partial \mu}. \tag{14}$$

The absolute correlations of the factors in $\mathcal{A}$ (except for the intercept) are λ, whereas the values are smaller than λ for the factors in $\mathcal{A}^c$.

The next predictor step extends $\hat{\boldsymbol{\beta}}$ as in equation (9), and, thus, the current correlations change. Denoting the vector of changes in correlation for a unit decrease in λ as $\mathbf{a}$,

$$\mathbf{c}(h) = \hat{\mathbf{c}} - h\mathbf{a} \tag{15}$$

$$= \hat{\mathbf{c}} - h\mathbf{X}' \hat{\mathbf{W}} \mathbf{X}_A (\mathbf{X}_A' \hat{\mathbf{W}} \mathbf{X}_A)^{-1} \operatorname{sgn}\begin{pmatrix} 0 \\ \hat{\beta} \end{pmatrix}, \tag{16}$$

where $h > 0$ is a given decrease in λ. For the factors in $\mathcal{A}$, the values of $\mathbf{a}$ are those of $\operatorname{sgn}\binom{0}{\hat{\beta}}$. To find the h with which any factor in $\mathcal{A}^c$ yields the same absolute correlation as those in $\mathcal{A}$, we solve the equations

$$|c_j(h)| = |\hat{c}_j - h a_j| = \lambda - h \qquad \text{for any } j \in \mathcal{A}^c. \tag{17}$$

The equations suggest an estimate of the step length in λ as

$$h = \min_{j \in \mathcal{A}^c}{}^{+} \left(\frac{\lambda - \hat{c}_j}{1 - a_j}, \frac{\lambda + \hat{c}_j}{1 + a_j} \right). \tag{18}$$

In addition, to check whether any variable in the active set reaches 0 before λ decreases by h, we solve the equations

$$\beta_j(\tilde{h}) = \hat{\beta}_j + \tilde{h} (\mathbf{X}_A' \hat{\mathbf{W}} \mathbf{X}_A)^{-1} \operatorname{sgn}\begin{pmatrix} 0 \\ \hat{\beta} \end{pmatrix} = 0 \qquad \text{for any } j \in \mathcal{A}. \tag{19}$$

If $0 < \tilde{h} < h$ for any $j \in \mathcal{A}$, we expect that the corresponding variable will be eliminated from the active set before any other variable joins it; therefore, $\tilde{h}$ rather than h is used as the next step length.

As a by-product of this step length approximation strategy, $\partial \boldsymbol{\beta} / \partial \lambda$ in equation (8) is computed. When fitting with high dimensional data, the active set changes with a small decrease in λ, and thus the role of predictor step as in equations (8)–(9) is not critical. However, we would still include predictor steps for the unusual cases of a large decrement in λ and, with the predictor step direction automatically computed, the remaining computations are trivial.

Letting the coefficient paths be piecewise linear with the knots placed where the active set changes is a reasonable simplification of the truth based on our experience (using both simulated and real data sets). If the smallest step length that modifies the active set were to be larger than

the value that we have estimated, the active set remains the same, even after the corrector step. If the true step length were smaller than expected and, thus, we missed the entering point of a new active variable by far, we would repeat a corrector step with an increased λ. (We estimate the increase in a manner that is analogous to expression (19).) Therefore, our path algorithm almost precisely detects the values of λ at which the active set changes, in the sense that we compute the exact coefficients at least once before their absolute values grow larger than δ (which is a small fixed quantity).

We can easily show that, in the case of Gaussian distribution with the identity link, the piecewise linear paths are exact. Because $\hat{\boldsymbol{\mu}} = \mathbf{X}\hat{\boldsymbol{\beta}}$, and $V_i = \mathrm{var}(y_i)$ is constant for $i = 1, \ldots, n$, $H(\boldsymbol{\beta}, \lambda)$ simplifies to

$$-\mathbf{X}'(\mathbf{y} - \boldsymbol{\mu}) + \lambda \,\mathrm{sgn}\begin{pmatrix} 0 \\ \beta \end{pmatrix}.$$

The step lengths are computed with no error; in addition, since the predictor steps yield the exact coefficient values, corrector steps are not necessary. In fact, the paths are identical to those of the lasso.

2.3. Degrees of freedom

We use the size of the active set as a measure of the degrees of freedom, which changes, not necessarily monotonically, along the solution paths, i.e.

$$\mathrm{df}(\lambda) = |\mathcal{A}(\lambda)|, \tag{20}$$

where $|\mathcal{A}(\lambda)|$ denotes the size of the active set corresponding to λ. This is based on $\mathrm{df}(\lambda) = E|\mathcal{A}(\lambda)|$, which holds in the case of the lasso. This remarkable formula was discovered by Efron *et al.* (2004) and improved by Zou and Hastie (2004); the effect of shrinking cancels the price that is paid in the variance for the aggressive searching for the best variable to include in the model. Here we present a heuristic justification for using equation (20) for GLMs in general, based on the results that were developed in Zou and Hastie (2004).

One can show that the estimates of $\boldsymbol{\beta}$ at the end of a corrector step solve a weighted lasso problem,

$$\min_{\boldsymbol{\beta}} \{(\mathbf{z} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}} \mathbf{W} (\mathbf{z} - \mathbf{X}\boldsymbol{\beta})\} + \lambda \|\boldsymbol{\beta}\|_1, \tag{21}$$

where the *working response* vector is defined as

$$\mathbf{z} = \boldsymbol{\eta} + (\mathbf{y} - \boldsymbol{\mu}) \frac{\partial \eta}{\partial \mu}. \tag{22}$$

The solution to problem (21) would be an appropriate fit for a linear model

$$\mathbf{z} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{23}$$

where $\boldsymbol{\varepsilon} \sim (\mathbf{0}, \mathbf{W}^{-1})$. This covariance is correct at the true values of $\boldsymbol{\eta}$ and $\boldsymbol{\mu}$, and can be defended asymptotically if appropriate assumptions are made. In fact, when $\lambda = 0$, model (23) leads directly to the standard asymptotic formulae and Gaussianity for the maximum likelihood estimates in the exponential family.

Under these heuristics, we apply *Stein's lemma* (Stein, 1981) to the transformed response $(\mathbf{W}^{1/2}\mathbf{z})$ so that its errors are homoscedastic. We refer readers to Zou and Hastie (2004) for the details of the application of the lemma. Simulations show that equation (20) approximates the degrees of freedom reasonably closely, although we omit the details here.

2.4. Adding a quadratic penalty

When some columns of $\mathbf{X}$ are strongly correlated, the coefficient estimates are highly unstable; the solutions might not be unique if some columns are linearly dependent or redundant in the sense that they do not satisfy the conditions for theorem 5 of Rosset *et al.* (2004). To overcome these situations, we propose to add a quadratic penalty term to the criterion, following the *elastic net* proposal of Zou and Hastie (2005), i.e. we compute the solution paths that satisfy

$$\hat{\boldsymbol{\beta}}(\lambda_1) = \arg\min_{\boldsymbol{\beta}}[-\log\{L(\mathbf{y};\boldsymbol{\beta})\} + \lambda_1\|\beta\|_1 + \frac{\lambda_2}{2}\|\beta\|_2^2], \tag{24}$$

where $\lambda_1 \in (0,\infty)$ and λ_2 is a fixed, small, positive constant. As a result, strong correlations between the features do not affect the stability of the fit. When the correlations are not strong, the effect of the quadratic penalty with a small λ_2 is negligible.

Assuming that all the elements of $\boldsymbol{\beta}$ are non-zero, if $\mathbf{X}$ does not have a full column rank, $\partial H(\boldsymbol{\beta},\lambda)/\partial\boldsymbol{\beta} = \mathbf{X}'\mathbf{W}\mathbf{X}$ is singular, where H is defined as in equation (6). By adding a quadratic penalty term, as in equation (24), we redefine H:

$$\tilde{H}(\boldsymbol{\beta},\lambda_1,\lambda_2) = -\mathbf{X}'\mathbf{W}(\mathbf{y}-\boldsymbol{\mu})\frac{\partial\eta}{\partial\mu} + \lambda_1\,\mathrm{sgn}\begin{pmatrix}0\\ \beta\end{pmatrix} + \lambda_2\begin{pmatrix}0\\ \beta\end{pmatrix}. \tag{25}$$

Accordingly, the following $\partial\tilde{H}/\partial\boldsymbol{\beta}$ is non-singular, in general, with any $\lambda_2 > 0$:

$$\frac{\partial\tilde{H}}{\partial\boldsymbol{\beta}} = \mathbf{X}'\mathbf{W}\mathbf{X} + \lambda_2\begin{pmatrix}0 & \mathbf{0}'\\ \mathbf{0} & I\end{pmatrix}. \tag{26}$$

Therefore, when λ_2 is fixed at a constant, and λ_1 varies in an open set, such that the current active set remains the same, a unique, continuous and differentiable function $\boldsymbol{\beta}(\lambda_1)$ satisfies $\tilde{H}\{\boldsymbol{\beta}(\lambda_1),\lambda_1,\lambda_2\} = 0$. This connection between the non-singularity and existence of a unique, continuous and differentiable coefficient path is based on the implicit function theorem (Munkres, 1991).

Zou and Hastie (2005) proposed elastic net regression, which added an L_2-norm penalty term to the criterion for the lasso. Zou and Hastie adjusted the values of both λ_1 and λ_2 so that variable selection and grouping effects were achieved simultaneously. For our purpose of handling inputs with strong correlations, we fixed λ_2 at a very small number, while changing the value of λ_1 for different amounts of regularization.

In the case of logistic regression, adding an L_2-penalty term is also helpful as it elegantly handles the separable data. Without the L_2-penalization, and if the data are separable by the predictors, $\|\hat{\beta}\|_1$ grows to ∞ as λ_1 approaches 0. Rosset *et al.* (2004) showed that the normalized coefficients $\hat{\beta}/\|\hat{\beta}\|_1$ converge to the L_1-margin maximizing separating hyperplane as λ_1 decreases to 0. In such cases, the fitted probabilities approach 0 or 1, and thus the maximum likelihood solutions are undefined. However, by restricting $\|\beta\|_2$ with any small amount of quadratic penalization, we let the coefficients converge to the L_2-penalized logistic regression solutions instead of ∞ as λ_1 approaches 0. As an alternative solution to the separation in logistic regression, one can apply the Jeffreys prior to the likelihood function as suggested in Firth (1993) and further demonstrated in Heinze and Schemper (2002).

3. Data analysis

In this section, we demonstrate our algorithm through a simulation and two real data sets: the *South African heart disease data* and *leukaemia cancer gene expression data*. Our examples focus on binary data: hence the logistic regression GLM.

3.1. Simulated data example

We simulated a data set of 100 observations with five variables and a binary response. Fig. 1 shows three sets of coefficient paths with respect to λ, with various selections of step sizes. In Fig. 1(a) the exact solutions were computed at the values of λ where the active set changed, and the solutions were connected in a piecewise linear manner. Fig. 1(b) shows the paths with exact solutions on a much finer grid of λ-values; we controlled the arc length to be less than 0.1 between any two adjacent values of λ. We observe the true curvature of the paths. Fig. 1(a) is a reasonable approximation of Fig. 1(b), especially because the active set is correctly specified at any value of λ. Fig. 1(c) shows the solution paths that we generated by using the boosted lasso algorithm by Zhao and Yu (2004). We used the negative (binomial) log-likelihood loss and the step size constant $\varepsilon = 0.1$, the maximum arc length of the previous GLM path; 60 and 58 steps were taken by the GLM path algorithm (Fig. 1(b)) and the boosted lasso algorithm (Fig. 1(c)) respectively. The boosted lasso solution paths are less smooth as the solutions oscillate around the real path as λ decreases.

3.2. South African heart disease data

The South African heart disease data set consists of nine different features of 462 samples as well as the responses indicating the presence of heart disease. The data set has also been used in Hastie *et al.* (2001) with a detailed description of the data. Using the disease–non-disease response variable, we can fit a logistic regression path.

Fig. 2(a) shows the exact set of paths; the coefficients were precisely computed at 300 different values of λ ranging from 81.9 to 0, with the constraint that every arc length be less than 0.01. The L_1-norm of the coefficients forms the x-axis, and the vertical breaks indicate where the active set is modified. Comparing this plot with Fig. 2(b), which we achieved in 13 steps rather than 300, we find that the two are almost identical. Our strategy to find the λ-values at which the active set changes resulted in an estimate of the values with reasonable accuracy. In addition, the exact paths are curvy but are almost indistinguishable from the piecewise linear version, justifying our simplification scheme. For both plots, the rightmost solutions corresponding to $\lambda = 0$ are the maximum likelihood estimates.

Fig. 2(c) illustrates the paths with respect to the steps. Two extra steps were needed between the knots at which the sixth and the seventh variable joined the active set. However, the step lengths in λ are tiny in this region; since the first approximation of λ that would change the active set was larger than the true value by only a small amount, λ decreased again by extremely small amounts. For most other steps, the subsequent λs that would modify the active set were accurately estimated on their first attempts.

We have proposed three different strategies for selecting the step sizes in λ in Section 2.2.4:

(a) fixing the step size Δ, $\Delta_k = \Delta$;
(b) fixing the arc length L, $\Delta_k = L/\|\partial \boldsymbol{\beta}/\partial\lambda\|_1$;
(c) estimating where the active set changes.

To verify that method (c) yields more accurate paths with a smaller number of steps, we present the following comparison. For the three methods, we counted the number of steps taken and computed the corresponding sum of squared errors in β, $\Sigma_{m=1}^{200}\|\hat{\beta}_{(m)} - \beta_{(m)}\|^2$. $\hat{\beta}_{(m)}$ and $\beta_{(m)}$ denote the coefficient estimates at the mth (out of 200 evenly spaced grid values in $\|\beta\|_1$) grid along the path, from the path generated by using a certain step length computation method and the exact path respectively.

As shown in the first row of Table 1, the first two strategies of selecting the step lengths, with a comparable number of steps, achieved much lower accuracy than the third. Furthermore, the

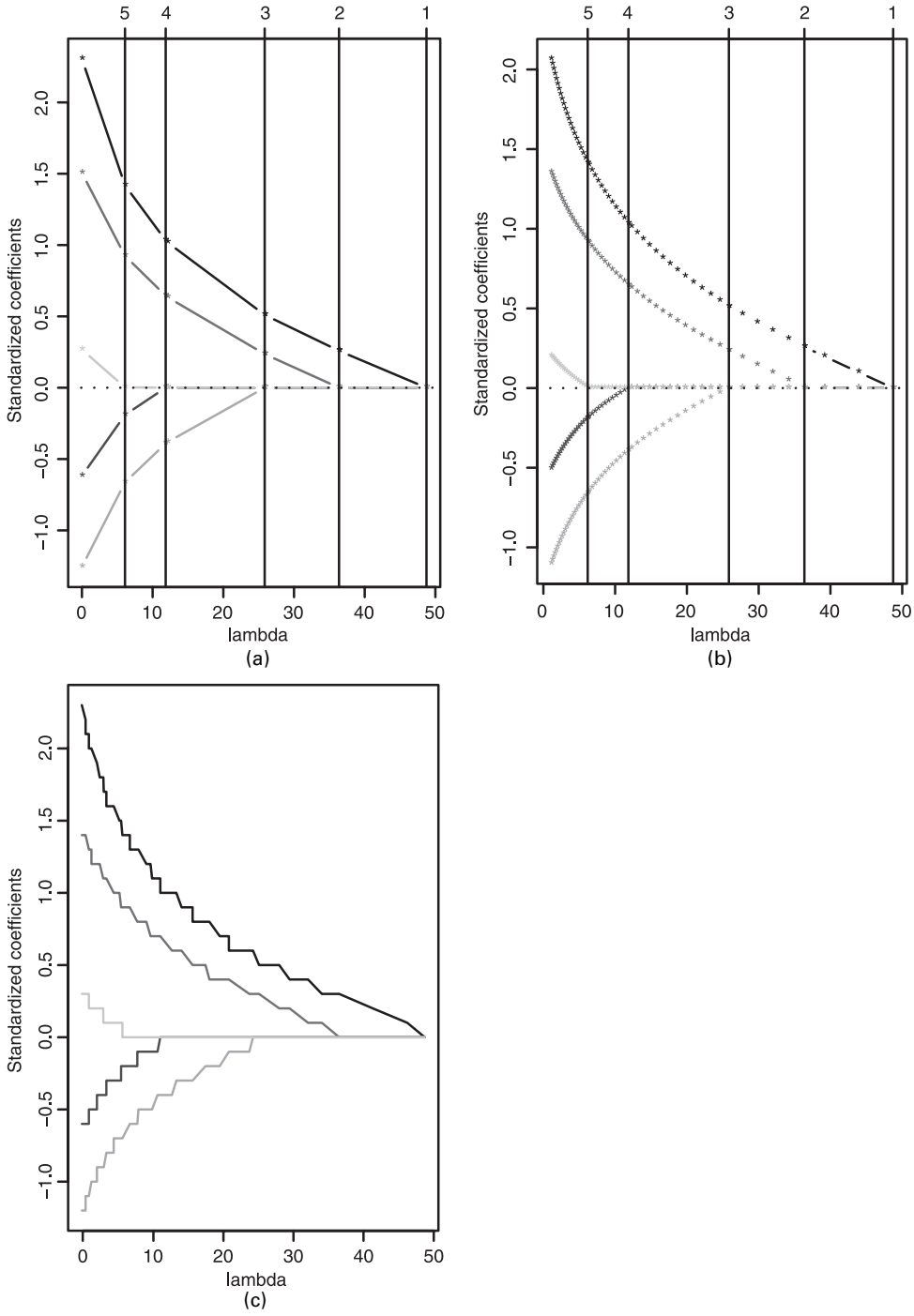

Fig. 1. Simulated data from Section 3.1: (a) exact solutions were computed at the values of λ where the active set changed, and the solutions were connected in a piecewise linear manner (GLM path); (b) GLM paths with exact solutions at much finer grids of λ (we controlled the arc length to be less than 0.1 between any two adjacent values of λ); (c) solution paths generated by using the boosted lasso algorithm, with the step size constant $\varepsilon = 0.1$

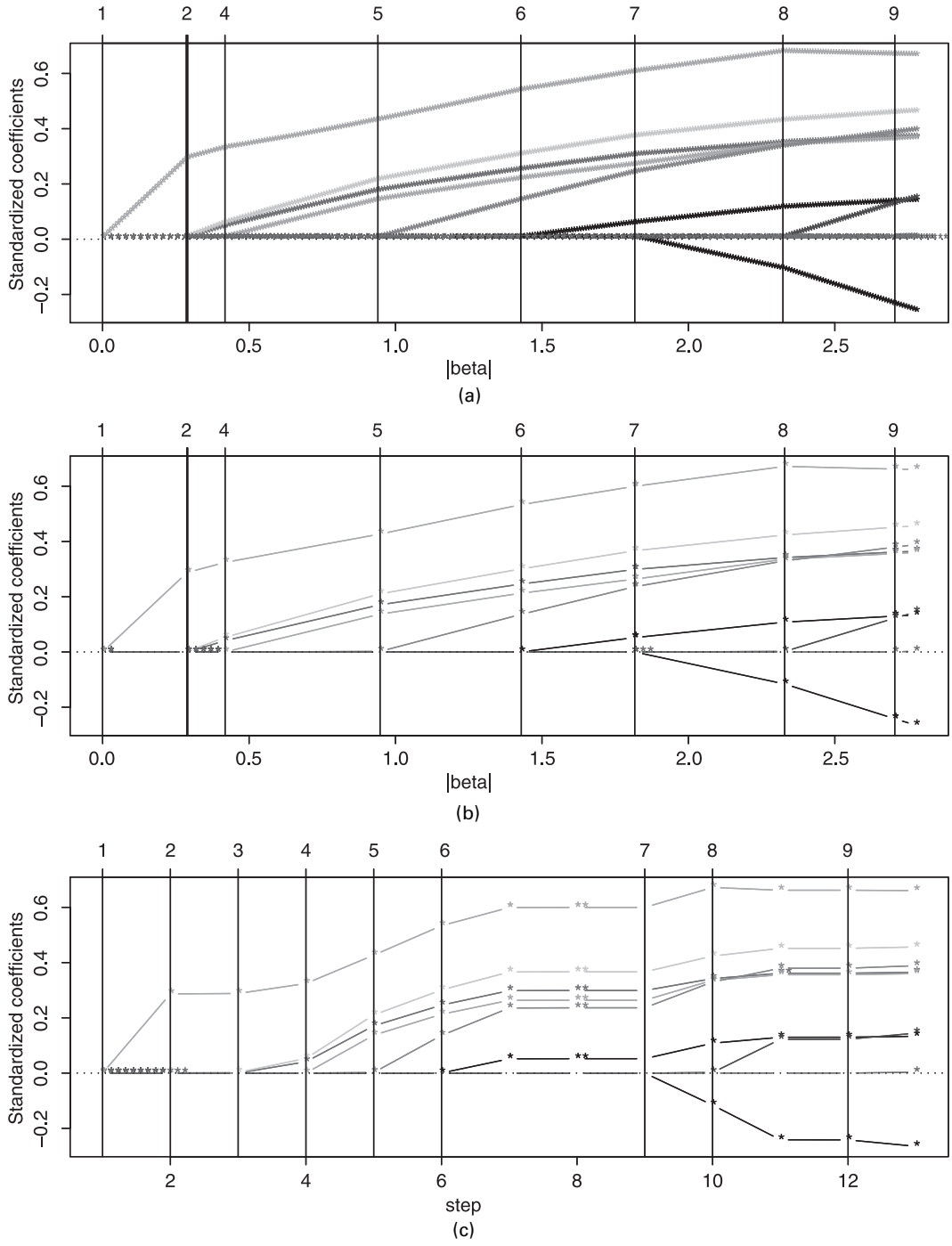

Fig. 2. Heart disease data from Section 3.2: (a) exact set of paths (the coefficients were precisely computed at 300 different grids of λ ranging from 81.9 to 0 with the constraint that every arc length be less than 0.01; the vertical breaks indicate where the active set is modified; the L_1-norm of the coefficients forms the x-axis); (b) exact set of paths achieved in 13 steps rather than 300 (comparing this with (a), we find that the two are almost identical); (c) paths as a function of step number, to illustrate how minor the corrections are

Table 1. Results from the South African heart disease data†

Results from method (a)			*Results from method (b)*			*Results from method (c)*	
Δ	*Number of steps*	*Error*	L	*Number of steps*	*Error*	*Number of steps*	*Error*
8	12	2.56×10^{-1}	0.23	11	1.01×10^{-1}	13	7.11×10^{-4}
1	83	2.04×10^{-1}	0.1	26	7.78×10^{-2}		
0.3	274	6.75×10^{-3}	0.02	142	2.28×10^{-2}		
0.15	547	7.16×10^{-5}	0.01	300	4.25×10^{-5}		

†As shown in the first row, the first two strategies of selecting the step lengths, with a comparable number of steps, achieved much lower accuracy than the third. The first two methods needed a few hundred steps to yield the same accuracy as the third method achieved in only 13 steps.

first two methods needed a few hundred steps to yield the same accuracy that the third method achieved in only 13 steps. Thus, method (c) provides accuracy and efficiency in addition to the information about where the junction points are located.

3.3. Leukaemia cancer gene expression data

The GLM path algorithm is suitable for data consisting of far more variables than the samples (so-called $p \gg n$ scenarios) because it successfully selects up to n variables along the regularization path regardless of the number of input variables. We demonstrate this use of our algorithm through a logistic regression applied to the leukaemia cancer gene expression data set by Golub *et al.* (1999). The data set contains the training and the test samples of sizes 38 and 34 respectively. For each sample, 7129 gene expression measurements and a label indicating the type of cancer (acute myeloid leukaemia or acute lymphoblastic leukaemia) are available.

Fig. 3(a) shows the coefficient paths that we achieved using the training data; the size of the active set cannot exceed the sample size at any segment of the paths (this fact is proved in Rosset *et al.* (2004)). The vertical line marks the chosen level of regularization (based on cross-validation), where 23 variables had non-zero coefficients. Fig. 3(b) illustrates the patterns of tenfold cross-validation and test errors. As indicated by the vertical line, we selected λ where the cross-validation error achieved the minimum.

Table 2 shows the errors and the number of variables that were used in the prediction. We also compared the performance with that of other methods that used the same data set in their literature. With a cross-validation error of 1/38 and a test error of 2/34, L_1 penalized logistic regression is comparable with or more accurate than other competing methods for analysis of this microarray data set.

Although we did not perform any preprocessing to filter from the original 7129 genes, automatic gene selection reduced the number of effective genes to 23. This set of genes included seven, 14, eight, 15 and five of the genes selected through L_2 penalized logistic regression (univariate ranking), L_2 penalized logistic regression (recursive feature elimination), the support vector machine (univariate ranking), the support vector machine recursive feature elimination and nearest shrunken centroid classification respectively. Each of these methods selected only a small proportion of the available genes, which were highly correlated within subgroups. As a result, the gene groups from different methods did not entirely overlap, but some of the genes were commonly significant across different models.

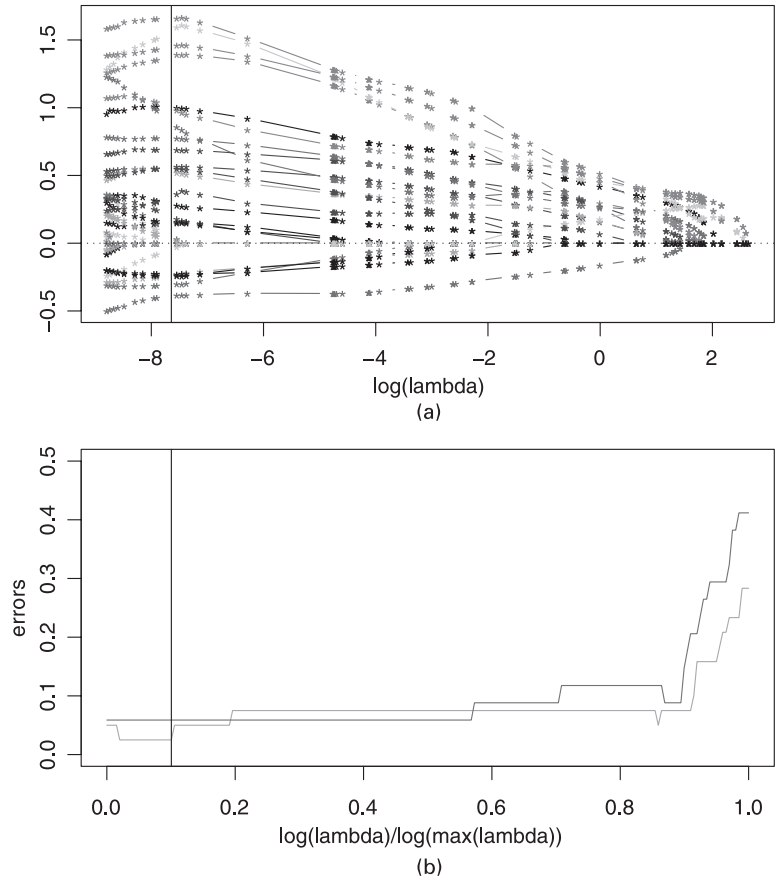

Fig. 3. Leukaemia data from Section 3.3: (a) coefficient paths achieved by using the training data (the size of the active set cannot exceed the sample size at any segment of the paths; |, chosen level of regularization (based on cross-validation), where 23 variables had non-zero coefficients); (b) patterns of tenfold cross-validation (·····) and test errors (———) (as indicated by the vertical line, we selected λ where the cross-validation error achieved the minimum

4. L_1-regularized Cox proportional hazards models

The path following method that we applied to the L_1-regularized GLM may also be used to generate other non-linear regularization paths. We illustrate an analogous implementation of the predictor–corrector method for drawing the L_1-regularization path for the Cox proportional hazards model (Cox, 1972). Tibshirani (1997) proposed fitting the Cox model with a penalty on the size of the L_1-norm of the coefficients. This shrinkage method computes the coefficients with a criterion similar to expression (2):

$$\hat{\beta}(\lambda) = \arg\min_{\beta}[-\log\{L(\mathbf{y};\beta)\} + \lambda\|\beta\|_1], \tag{27}$$

where L denotes the partial likelihood. We formulate the entire coefficient paths $\{\hat{\beta}(\lambda) : 0 < \lambda < \lambda_{\max}\}$, where $\lambda_{\max}$ is the largest λ that makes $\hat{\beta}(\lambda)$ non-zero, through the predictor–corrector scheme. As a result of L_1-penalization, the solutions are sparse; thus, the active set changes along with λ.

Table 2. Results from the leukaemia data†

Method	*Cross-validation error*	*Test error*	*Number of genes used*
L_1 penalized logistic regression	1/38	2/34	23
L_2 penalized logistic regression (univariate ranking) (Zhu and Hastie, 2004)	2/38	3/34	16
L_2 penalized logistic regression (recursive feature elimination) (Zhu and Hastie, 2004)	2/38	1/34	26
Support vector machine (univariate ranking) (Zhu and Hastie, 2004)	2/38	3/34	22
Support vector machine (recursive feature elimination) (Zhu and Hastie, 2004)	2/38	1/34	31
Nearest shrunken centroid classification (Tibshirani *et al.*, 2002)	1/38	2/34	21

†With a cross-validation error of 1/38 and a test error of 2/34, L_1 penalized logistic regression is comparable with or more accurate than other competing methods for analysis of this microarray data set. Although we did not perform any preprocessing to filter from the original 7129 genes, automatic gene selection reduced the number of effective genes to 23.

4.1. Method

Let $\{(\mathbf{x}_i, y_i, \delta_i) : \mathbf{x}_i \in \mathcal{R}^p, y_i \in \mathcal{R}_+, \delta_i \in \{0, 1\}, i = 1, \ldots, n\}$ be n triples of p factors, a response indicating the survival time and a binary variable $\delta_i = 1$ for complete (died) observations and $\delta_i = 0$ for right-censored patients. On the basis of criterion (27), we find the coefficients that minimize the following objective function for each λ:

$$l(\beta, \lambda) = -\sum_{i=1}^{n} \delta_i \beta' x_i + \sum_{i=1}^{n} \delta_i \log\Big\{ \sum_{j \in R_i} \exp(\beta' x_j) \Big\} + \lambda \|\beta\|_1, \tag{28}$$

where R_i is the risk set at time y_i. To compute the coefficients, we solve $H(\beta, \lambda) = 0$ for β, where H is defined as follows using only the current non-zero components of β:

$$H(\beta, \lambda) = \frac{\partial l}{\partial \beta} = -\sum_{i=1}^{n} \delta_i x_i + \sum_{i=1}^{n} \delta_i \sum_{j \in R_i} w_{ij} x_j + \lambda \operatorname{sgn}(\beta), \tag{29}$$

where $w_{ij} = \exp(\beta' x_j) / \Sigma_{m \in R_i} \exp(\beta' x_m)$.

We refer the readers to Appendix B for further details of the procedure.

4.2. Real data example

We demonstrate the L_1-regularization path algorithm for the Cox model by using the heart transplant survival data that were introduced in Crowley and Hu (1977). The data set consists of 172 samples with their survival time and censoring information, as well as these four features:

(a) age (age – 48 years),
(b) year (year of acceptance, in years after November 1st, 1967),
(c) surgery (prior bypass surgery; 1, if yes) and
(d) transplant (received transplant; 1, if yes).

In Fig. 4(a), the coefficients were computed at fine grids of λ, whereas, in Fig. 4(b), the solutions were computed only when the active set was expected to change. Similarly to the GLM

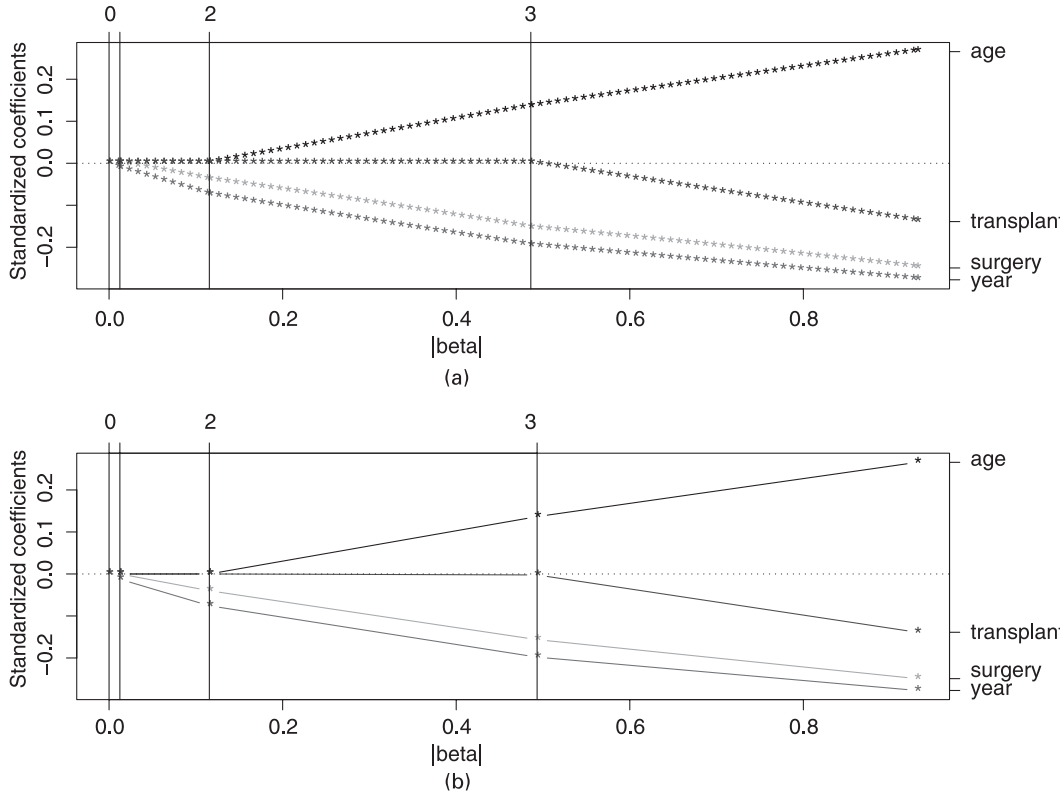

Fig. 4. Survival data from Section 4.2: (a) coefficient paths computed at fine grids of λ (similar to the GLM path examples, the exact coefficient paths are almost piecewise linear); (b) solutions computed only when the active set was expected to change (it is difficult to distinguish the two versions generated by different step sizes of λ)

path examples, the exact coefficient paths that are shown in Fig. 4(a) are almost piecewise linear; it is difficult to distinguish the two versions that were generated by different step sizes in λ.

5. Discussion

In this paper, we have introduced a path following algorithm to fit GLMs with L_1-regularization. As applied to regression (Tibshirani, 1996, 1997) and classification methods (Genkin *et al.*, 2004; Shevade and Keerthi, 2003; Zhu *et al.*, 2003), penalizing the size of the L_1-norm of the coefficients is useful because it accompanies variable selection. This strategy has provided us with a much smoother feature selection mechanism than the forward stepwise process.

Although the regularization parameter (λ in our case) influences the prediction performance in the aforementioned models considerably, determining the parameter can be troublesome or can demand heavy computation. The GLM path algorithm facilitates model selection by implementing the predictor–corrector method and finding the entire regularization path. Even with large intervals in λ, the predictor steps provide the subsequent corrector steps with reasonable estimates (starting values); therefore, the intervals can be wide without increasing the computations by a large number, as long as the paths can be assumed to be approximately linear within the intervals. Having generated the path, we estimate the globally optimal shrinkage level by cross-validating on the ratio $\lambda/\lambda_{\max}$ or $\log(\lambda)/\log(\lambda_{\max})$. We compute the cross-validated

loss at fixed values of the ratio rather than the values of λ because $\lambda_{\max}$ varies for every fold and, thus, a certain value of λ may correspond to different magnitude of shrinkage within each fold.

In Section 3.2, we proposed three different methods to determine the step lengths in λ and emphasized the efficiency and accuracy of the strategy of finding the transition points. One may suggest a more naïve approach of preselecting certain values of λ and generating the coefficient paths by connecting the solutions to those grids. However, computing the exact solutions at the values of λ where the active set changes ensures that we correctly specify the order in which variables are selected.

We can extend the use of the predictor–corrector scheme by generalizing the *loss plus penalty* function to any convex and almost differentiable functions. For example, we can find the entire regularization path for the Cox proportional hazards model with L_1-penalization, as described in Section 4. Rosset and Zhu (2003) illustrated sufficient conditions for the regularized solution paths to be piecewise linear. Just as the solution paths for Gaussian distributions were computed with no error through the predictor–corrector method, so any other piecewise linear solution paths can be computed exactly by applying the same strategy.

The path following algorithms for GLMs and Cox proportional hazards models have been implemented in the contributed R package `glmpath` which is available from the Comprehensive R Archive Network at `http://cran.r-projet.org/src/contrib/Descriptions/glmpath.html`.

Acknowledgements

We thank Michael Saunders of the Systems Optimization Laboratory, Stanford University, for helpful discussions, and for providing the solver that we used in the corrector steps of our algorithms. We thank Robert Tibshirani for helpful comments and suggestions. We are also grateful to the reviewers for valuable inputs. Trevor Hastie was partially supported by grant DMS-0505676 from the National Science Foundation and grant 2R01 CA 72028-07 from the National Institutes of Health.

Appendix A: Proofs

A.1. Proof of lemma 1

Minimizing expression (5) is equivalent to minimizing

$$-\sum_{i=1}^{n}[y_i\,\theta(\boldsymbol{\beta})_i - b\{\theta(\boldsymbol{\beta})_i\}] + \sum_{j=1}^{p}\{\lambda(\beta_j^+ + \beta_j^-) - \lambda_j^+\beta_j^+ - \lambda_j^-\beta_j^-\}, \tag{30}$$

where $\beta = \beta^+ + \beta^-$, $\beta_j^+\beta_j^- = 0$, $\beta_j^+, \beta_j^- \geqslant 0$ and $\lambda_j^+, \lambda_j^- \geqslant 0$, $\forall j = 1, \ldots, p$.

The Karush–Kuhn–Tucker (KKT) optimality conditions for this equivalent criterion are

$$-\mathbf{x}_j'\hat{\mathbf{W}}(\mathbf{y} - \hat{\boldsymbol{\mu}})\frac{\partial\eta}{\partial\mu} + \lambda - \lambda_j^+ = 0, \tag{31}$$

$$-\mathbf{x}_j'\hat{\mathbf{W}}(\mathbf{y} - \hat{\boldsymbol{\mu}})\frac{\partial\eta}{\partial\mu} + \lambda - \lambda_j^- = 0, \tag{32}$$

$$\lambda_j^+\hat{\beta}_j^+ = 0, \tag{33}$$

$$\lambda_j^-\hat{\beta}_j^- = 0, \qquad \forall j = 1, \ldots, p. \tag{34}$$

The KKT conditions imply that

$$\left|\mathbf{x}_j'\hat{\mathbf{W}}(\mathbf{y}-\hat{\boldsymbol{\mu}})\frac{\partial\eta}{\partial\mu}\right| < \lambda \Rightarrow \hat{\beta}_j = 0 \qquad \text{for } j=1,\ldots,p. \tag{35}$$

When $\hat{\beta}_j = 0$ for all $j=1,\ldots,p$, the KKT conditions again imply

$$\mathbf{1}'\hat{\mathbf{W}}(\mathbf{y}-\hat{\boldsymbol{\mu}})\frac{\partial\eta}{\partial\mu} = 0, \tag{36}$$

which, in turn, yields $\hat{\boldsymbol{\mu}} = \bar{y}\mathbf{1} = g^{-1}(\hat{\beta}_0)\mathbf{1}$.

A.2. Proof of theorem 1

Since

$$\frac{\partial\boldsymbol{\beta}}{\partial\lambda} = -(\mathbf{X}_A'\mathbf{W}_k\mathbf{X}_A)^{-1}\,\mathrm{sgn}\begin{pmatrix}0\\ \hat{\beta}^k\end{pmatrix}$$

is continuously differentiable with respect to $\lambda \in (\lambda_{k+1}, \lambda_k]$,

$$\hat{\boldsymbol{\beta}}^{k+1} = \hat{\boldsymbol{\beta}}^k - h_k\frac{\partial\boldsymbol{\beta}}{\partial\lambda}\bigg|_{\lambda_k} + O(h_k^2) \tag{37}$$

$$= \hat{\boldsymbol{\beta}}^{k+} + O(h_k^2), \tag{38}$$

from which the conclusion follows.

A.3. Proof of theorem 2

Since $\partial\boldsymbol{\beta}/\partial\lambda$ is continuously differentiable with respect to $\lambda \in (\lambda_{k+1}, \lambda_k]$, the following equations hold:

$$\hat{\boldsymbol{\beta}}(\lambda-\alpha h_k) = \hat{\boldsymbol{\beta}}^k - \alpha h_k\frac{\hat{\boldsymbol{\beta}}^{k+1}-\hat{\boldsymbol{\beta}}^k}{-h_k} \tag{39}$$

$$= \hat{\boldsymbol{\beta}}^k - \alpha h_k\frac{\partial\boldsymbol{\beta}}{\partial\lambda}\bigg|_{\lambda_k} + O(h_k^2), \tag{40}$$

and similarly the true solution at $\lambda = \lambda_k - \alpha h_k$ is

$$\boldsymbol{\beta}(\lambda-\alpha h_k) = \hat{\boldsymbol{\beta}}^k - \alpha h_k\frac{\partial\boldsymbol{\beta}}{\partial\lambda}\bigg|_{\lambda_k} + O(h_k^2). \tag{41}$$

The conclusion follows directly from the above equations.

A.4. Proof of lemma 2

The KKT optimality conditions (31)–(34) imply that

$$\hat{\beta}_j \neq 0 \Rightarrow \left|\mathbf{x}_j'\hat{\mathbf{W}}(\mathbf{y}-\hat{\boldsymbol{\mu}})\frac{\partial\eta}{\partial\mu}\right| = \lambda. \tag{42}$$

This condition, combined with conditions (7) and (35), proves the argument.

Appendix B: L_1-regularization path algorithm for the Cox model

Here we describe details of the L_1-regularization path algorithm for the Cox model, which was briefly introduced in Section 4. We use the same notation as presented in Section 4.

The Cox model with L_1-penalization finds the coefficients that minimize the objective function:

$$l(\beta, \lambda) = -\sum_{i=1}^{n} \delta_i \beta' x_i + \sum_{i=1}^{n} \delta_i \log \Big\{ \sum_{j \in R_i} \exp(\beta' x_j) \Big\} + \lambda \|\beta\|_1. \tag{43}$$

The first and the second derivatives of l with respect to β are

$$\frac{\partial l}{\partial \beta} = H(\beta, \lambda) = -\sum_{i=1}^{n} \delta_i x_i + \sum_{i=1}^{n} \delta_i \sum_{j \in R_i} w_{ij} x_j + \lambda \operatorname{sgn}(\beta), \tag{44}$$

$$\frac{\partial^2 l}{\partial \beta \, \partial \beta'} = \frac{\partial H}{\partial \beta} = \sum_{i=1}^{n} \partial_i \Big\{ \sum_{j \in R_i} x_j x_j' w_{ij} - \Big(\sum_{j \in R_i} x_j w_{ij} \Big) \Big(\sum_{j \in R_i} x_j' w_{ij} \Big) \Big\} \tag{45}$$

$$= \mathbf{X}' \mathbf{A} \mathbf{X}, \tag{46}$$

where $w_{ij} = \exp(\beta' x_j) / \Sigma_{m \in R_i} \exp(\beta' x_m)$, and $\mathbf{A} = \partial l^2 / \partial \eta \, \partial \eta'$ with $\eta = \mathbf{X}\beta$.

If $\beta_j = 0$ for $j = 1, \ldots, p$, then $w_{ij} = 1/|R_i|$ for $i = 1, \ldots, n$ and $j = 1, \ldots, p$, and

$$\frac{\partial l}{\partial \beta} = -\sum_{i=1}^{n} \delta_i \Big(x_i - \frac{1}{|R_i|} \sum_{j \in R_i} x_j \Big). \tag{47}$$

$\hat{\beta}_j = 0$ for all j if $\lambda > \max_{j \in \{1, \ldots, p\}} |\partial l / \partial \beta_j|$. As λ is decreased further, an iterative procedure begins; variables enter the active set, beginning with $j_0 = \arg\max_j |\partial l / \partial \beta_j|$. The four steps of an iteration are as follows.

(a) Predictor step: in the kth predictor step, $\beta(\lambda_{k+1})$ is approximated as in equation (8), with

$$\frac{\partial \beta}{\partial \lambda} = -\Big(\frac{\partial H}{\partial \beta} \Big)^{-1} \frac{\partial H}{\partial \lambda} = -(\mathbf{X}_A' \mathbf{A} \mathbf{X}_A)^{-1} \operatorname{sgn}(\beta). \tag{48}$$

$\mathbf{X}_A$ contains the columns of $\mathbf{X}$ for the current active variables.

(b) Corrector step: in the kth corrector step, we compute the exact solution $\beta(\lambda_{k+1})$ by using the approximation from the previous predictor step as the initial value.

(c) Active set: denoting the correlation between the factors and the current residual as $\hat{\mathbf{c}}$,

$$\hat{\mathbf{c}} = \sum_{i=1}^{n} \delta_i x_i - \sum_{i=1}^{n} \delta_i \sum_{j \in R_i} w_{ij} x_j. \tag{49}$$

After each corrector step, if $|\hat{\mathbf{c}}_l| > \lambda$ for any $l \in \mathcal{A}^c$, we augment the active set by adding x_l. Corrector steps are repeated until the active set is not augmented further. If $\hat{\beta}_l = 0$ for any $l \in \mathcal{A}$, we eliminate x_l from the active set.

(d) Step length: if $\lambda = 0$, the algorithm stops. If $\lambda > 0$, we approximate the smallest decrement in λ with which the active set will be modified. As λ is decreased by h, the approximated change in the current correlation (49) is

$$\mathbf{c}(h) = \hat{\mathbf{c}} - h \mathbf{X}' \mathbf{A} \mathbf{X}_A (\mathbf{X}_A \mathbf{A} \mathbf{X}_A)^{-1} \operatorname{sgn}(\hat{\beta}). \tag{50}$$

On the basis of equation (50), we approximate the next largest λ at which the active set will be augmented or reduced.

References

Allgower, E. and Georg, K. (1990) *Numerical Continuation Methods*. Berlin: Springer.

Cox, D. R. (1972) Regression models and life-tables (with discussion). *J. R. Statist. Soc.* B, **34**, 187–220.

Crowley, J. and Hu, M. (1977) Covariance analysis of heart transplant survival data. *J. Am. Statist. Ass.*, **72**, 27–36.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004) Least angle regression. *Ann. Statist.*, **32**, 407–499.

Firth, D. (1993) Bias reduction of maximum likelihood estimates. *Biometrika*, **80**, 27–38.

Garcia, C. and Zangwill, W. (1981) *Pathways to Solutions, Fixed Points and Equilibria*. Englewood Cliffs: Prentice Hall.

Genkin, A., Lewis, D. and Madigan, D. (2004) Large-scale Bayesian logistic regression for text categorization. *Technical Report*. Rutgers University, Piscataway.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J.,

Caligiuri, M., Bloomfield, C. and Lander, E. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.
Hastie, T., Rosset, S., Tibshirani, R. and Zhu, J. (2004) The entire regularization path for the support vector machine. *J. Mach. Learn. Res.*, **5**, 1391–1415.
Hastie, T., Tibshirani, R. and Friedman, J. (2001) *Elements of Statistical Learning; Data Mining, Inference, and Prediction*. New York: Springer.
Heinze, G. and Schemper, M. (2002) A solution to the problem of separation in logistic regression. *Statist. Med.*, **21**, 2409–2419.
Lokhorst, J. (1999) The lasso and generalised linear models. *Technical Report*. University of Adelaide, Adelaide.
McCullagh, P. and Nelder, J. A. (1989) *Generalized Linear Models*, 2nd edn. London: Chapman and Hall.
Munkres, J. (1991) *Analysis on Manifolds*. Reading: Addison-Wesley.
Osborne, M., Presnell, B. and Turlach, B. (2000) A new approach to variable selection in least squares problems. *IMA J. Numer. Anal.*, **20**, 389–403.
Rosset, S. (2004) Tracking curved regularized optimization solution paths. In *Neural Information Processing Systems*. Cambridge: MIT Press.
Rosset, S. and Zhu, J. (2003) Piecewise linear regularized solution paths. *Technical Report*. Stanford University, Stanford.
Rosset, S., Zhu, J. and Hastie, T. (2004) Boosting as a regularized path to a maximum margin classifier. *J. Mach. Learn. Res.*, **5**, 941–973.
Shevade, S. and Keerthi, S. (2003) A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, **19**, 2246–2253.
Stein, C. (1981) Estimation of the mean of a multivariate normal distribution. *Ann. Statist.*, **9**, 1135–1151.
Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *J. R. Statist. Soc.* B, **58**, 267–288.
Tibshirani, R. (1997) The lasso method for variable selection in the cox model. *Statist. Med.*, **16**, 385–395.
Tibshirani, R., Hastie, T., Narasimhan, B. and Chu, G. (2002) Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Natn. Acad. Sci. USA*, **99**, 6567–6572.
Zhao, P. and Yu, B. (2004) Boosted lasso. *Technical Report*. University of California at Berkeley, Berkeley.
Zhu, J. and Hastie, T. (2004) Classification of gene microarrays by penalized logistic regression. *Biostatistics*, **46**, 505–510.
Zhu, J., Rosset, S., Hastie, T. and Tibshirani, R. (2003) 1-norm support vector machines. In *Neural Information Processing Systems*. Cambridge: MIT Press.
Zou, H. and Hastie, T. (2004) On the "degrees of freedom" of the lasso. *Technical Report*. Stanford University, Stanford.
Zou, H. and Hastie, T. (2005) Regularization and variable selection via the elastic net. *J. R. Statist. Soc.* B, **67**, 301–320.