Efficient Global Approximation of Generalized Nonlinear ℓ_1-Regularized Solution Paths and Its Applications

Ming YUAN and Hui ZOU

We consider efficient construction of nonlinear solution paths for general ℓ_1-regularization. Unlike the existing methods that incrementally build the solution path through a combination of local linear approximation and recalibration, we propose an efficient global approximation to the whole solution path. With the loss function approximated by a quadratic spline, we show that the solution path can be computed using a generalized Lars algorithm. The proposed methodology avoids high-dimensional numerical optimization and thus provides faster and more stable computation. The methodology also can be easily extended to more general regularization framework. We illustrate such flexibility with several examples, including a generalization of the elastic net and a new method that effectively exploits the so-called "support vectors" in kernel logistic regression.

KEY WORDS: ℓ_1-regularization; LARS; LASSO; Solution path; Support vector pursuit.

1. INTRODUCTION

In a general predictive modeling framework, a response variable Y is related to a p-dimensional explanatory variable $\mathbf{X} \equiv (X_1, \ldots, X_p)'$ through

$$\eta_0(\mathbf{X}) = \beta_0 + \mathbf{X}'\beta, \tag{1}$$

where $\beta = (\beta_1, \ldots, \beta_p)'$ is a p-dimensional unknown coefficient vector. When $\mathbf{X}$ is observable, the value of $\eta_0(\mathbf{X})$ or its transformation can be used as a predictor of Y. The goal, therefore, is to retrieve $\eta_0(\cdot)$ or, equivalently, β_0 and β, given a set of training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ consisting of n independent copies of $(\mathbf{X}, Y)$. A commonly used approach to estimating η_0 is regularized empirical risk minimization. Let $L(Y, \eta(\mathbf{X}))$ be a loss function such that its expectation with respect to the joint distribution of $(\mathbf{X}, Y)$, $E[L(Y, \eta(\mathbf{X}))]$, is minimized at $\eta_0(\cdot)$. Then η_0 can be estimated by

$$\widehat{\eta} = \arg\min_{\eta} [L_n(\eta) + \lambda J(\eta)], \tag{2}$$

where

$$L_n(\eta) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \eta(\mathbf{x}_i)) \tag{3}$$

and $J(\cdot)$ is a penalty function. For brevity, we write $L_n(\eta)$ and $L_n(\beta_0, \beta)$ interchangeably in what follows. The penalty $J(\cdot)$ is often defined through the coefficient vector β, that is, $J(\eta) \equiv J(\beta)$. Many popular prediction methods can be formulated using this framework; for example, in ridge regression, $L(Y, \eta(\mathbf{X})) = [Y - \eta(\mathbf{X})]^2$ and $J(\beta) = \|\beta\|_{\ell_2}^2 = \sum_{j=1}^{p} \beta_j^2$.

When p is large, it is often the case that some of the predictors have only marginal influence on the response. Effectively removing these insignificant explanatory variables could drastically improve estimation accuracy and enhance model interpretability. An extremely successful approach for exploiting such sparsity is to use ℓ_1-regularization in the empirical risk minimization,

$$J(\beta) = \|\beta\|_{\ell_1} = \sum_{j=1}^{p} |\beta_j|. \tag{4}$$

In the case of multiple linear regression, this becomes the popular Lasso (Tibshirani 1996). Under mild regularity conditions on the loss function, the ℓ_1-norm penalty induces sparsity in the estimated regression coefficient and excludes the insignificant explanatory variables through continuous shrinkage. For some particular types of loss functions, the ℓ_1-regularization methods are efficient to compute; for example, in multiple linear regression, Efron et al. (2004) proposed the Lars algorithm for computing the entire solution paths of the Lasso. Osborne, Presnell, and Turlach (2000) also considered a similar homotopy algorithm for solving the Lasso. Rosset and Zhu (2007) investigated a class of ℓ_1-penalized problems with piecewise linear solution paths. Lars-type path-following algorithms also have been used to compute solution paths of the Huberized Lasso (Rosset and Zhu 2007), the ℓ_1-penalized support vector machine (Zhu et al. 2003; Hastie et al. 2005), and the ℓ_1-penalized quantile regression (Li and Zhu 2008).

There are, however, many interesting ℓ_1-penalized empirical risk minimization problems whose solution paths are not piecewise linear and for which the aforementioned algorithms cannot be directly applied. One common example is the ℓ_1-penalized logistic regression. Rosset (2004) suggested a general algorithm for following "curved" regularized solution paths. His algorithm iteratively changes the regularization parameter and updates the coefficient estimate by a Newton iteration. Zhao and Yu (2007) proposed the so-called "boosted Lasso," that corrects the forward-stagewise boosting algorithm by allowing backward steps whenever a step in forward-stagewise boosting fitting deviates from that of the Lasso. Park and Hastie (2007) proposed a predictor–corrector algorithm that computes the entire solution path of ℓ_1-penalized generalized linear models. All

Ming Yuan is Associate Professor, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205 (E-mail: *myuan@isye.gatech.edu*). Hui Zou is Associate Professor, School of Statistics, University of Minnesota, Minneapolis, MN 55455 (E-mail: *hzou@stat.umn.edu*). Yuan's research was supported in part by National Science Foundation (NSF) grant DMS-0706724. Zou's research was supported in part by NSF grant DMS-0706733.

Journal of the American Statistical Association
December 2009, Vol. 104, No. 488, Theory and Methods
DOI: 10.1198/jasa.2009.tm08287

of these approximate solution paths are built in an incremental fashion. The approximation error is carefully monitored and controlled locally at each stage. Whenever necessary, the solution path is recalibrated to control the elevated approximation error. Because the approximating paths may comprise numerous steps, the cumulative cost of locally controlling the approximation error could be expensive. Furthermore, recalibration involves solving a high-dimensional nonlinear optimization problem. This is often done through iterative algorithms, which may run into numerical instability.

In this article we propose a global approach to efficiently approximating the nonlinear ℓ_1-regularization solution paths. In contrast to the existing work, our algorithm controls the approximation error globally, thereby allowing us to reduce the computational cost incurred in iterative local approximation methods. The proposed approach starts with a quadratic spline approximation to the loss function, which most often can be made arbitrarily accurate. For the quadratic spline loss, a generalized Lars-type algorithm is devised to compute the exact solution path, which we call the efficient global approximation (EGA) path. We show that the EGA path can well approximate the original nonlinear ℓ_1-regularization solution path, and that its approximation error is controlled by the approximation accuracy of the quadratic spline to the loss function. As opposed to the aforementioned local methods, our approach does not require numerical optimization. Once the approximate loss is adopted, the whole solution path can be computed explicitly and efficiently.

This article is organized as follows. In Section 2 we describe the EGA path methodology, derive the algorithm for computing the EGA paths, and show that these EGA paths can be arbitrarily close to the original nonlinear paths if a sufficiently good approximation to the loss function is adopted. In Section 3 we extend the methodology to construct solution paths for several more general regularization methods. In particular, we apply our algorithm to solve a generalization of the elastic net (Zou and Hastie 2005) and the support-vector pursuit problem in kernel logistic regression.

2. GLOBAL APPROXIMATION OF ℓ_1–REGULARIZATION PATH

In practice, ℓ_1-regularized empirical risk minimization proceeds in two steps. First, a solution path indexed by λ is built. Then the final model is selected on the solution path by cross-validation or using a criterion such as the Akaike information criterion (AIC) or the Bayes information criterion (BIC). In general, the solution path must be approximated by evaluating $\beta(\lambda)$ for a fine grid of tuning parameters, and there is a trade-off between approximation accuracy and computational cost in determining how fine a grid of tuning parameters should be considered. Park and Hastie (2007) proposed a strategy to alleviate such problems by recognizing that the solution path may be approximated reasonably well in certain regions with only few tuning parameters. An even more challenging problem is the numerical instability in calculating $\beta(\lambda)$ for a given λ. This is routinely done through numerical optimization, which is iterative in nature. Even when the objective function is strictly convex, algorithms for doing this can be unstable with high-dimensional data, as we demonstrate in Section 2.4.

Here we tackle the problem from a different angle. Instead of approximating $\beta(\lambda)$ for a given λ, we attempt to approximate the vector-valued function $\beta(\cdot)$. Recall that

$$(\beta_0(\lambda), \beta(\lambda)) = \arg\min\left[\frac{1}{n}\sum_{i=1}^{n} L(y_i, \eta(\mathbf{x}_i)) + \lambda\sum_{j=1}^{p}|\beta_j|\right]. \quad (5)$$

We propose to approximate $(\beta_0(\lambda), \beta(\lambda))$ by

$$(\tilde{\beta}_0(\lambda), \tilde{\beta}(\lambda)) = \arg\min\left[\frac{1}{n}\sum_{i=1}^{n} \tilde{L}(y_i, \eta(\mathbf{x}_i)) + \lambda\sum_{j=1}^{p}|\beta_j|\right], \quad (6)$$

where $\tilde{L}$ is an approximation to L. The choice of $\tilde{L}$ should ensure that the solution path $(\tilde{\beta}_0(\cdot), \tilde{\beta}(\cdot))$ closely approximates the original solution path $(\beta_0(\cdot), \beta(\cdot))$ and also is easy to compute. In particular, we consider approximating L by a quadratic spline and show that such approximation enjoys both properties.

2.1 Approximate Loss Function

Most common loss functions are twice differentiable with respect to η. For brevity, in this article we focus on these loss functions; however, the discussion can be easily extended to the more general situations as long as L is convex with respect to η. When L is twice differentiable, it is desirable for its approximation also to enjoy such properties. Toward this end, we consider approximating L using a quadratic spline,

$$\tilde{L}(Y, \eta) = a_0(Y)\eta(\mathbf{X})^2 + b_0(Y)\eta(\mathbf{X}) + c_0(Y) + \sum_{j=1}^{M} d_j(Y)(\eta(\mathbf{X}) - \kappa_j(Y))_+^2, \quad (7)$$

where $(x)_+$ represents the positive part of x and $\kappa_1 < \cdots < \kappa_M$ are the so-called "knots." Various loss functions can be written in the form of (7); popular examples include quadratic loss, $L(Y, \eta) = (Y - \eta)^2$, and squared hinge loss, $L(Y, \eta) = (1 - Y\eta)_+^2$, among others. More generally, good quadratic spline approximations can be obtained for a large class of loss functions; for example, when $\partial^2 L/\partial\eta^2$ satisfies uniform Lipschitz condition of order $\alpha \geq 0$, in that

$$\sup_Y \left|\left.\frac{\partial^2 L}{\partial\eta^2}\right|_{(Y,\eta_1)} - \left.\frac{\partial^2 L}{\partial\eta^2}\right|_{(Y,\eta_2)}\right| \leq C_1|\eta_1 - \eta_2|^\alpha \quad (8)$$

for any $|\eta_1|, |\eta_2| \leq C_2$ and some constants $C_1, C_2 > 0$, from a Taylor expansion, it can be derived that there exists a quadratic spline, $\tilde{L}$, of form (7) so that

$$\sup_{|\eta|\leq C_2} |L(Y, \eta) - \tilde{L}(Y, \eta)| = O\big(M^{-(2+\alpha)}\big). \quad (9)$$

We now show that the closeness between L and $\tilde{L}$ generally entails the closeness between the corresponding solution paths, and thus the solution path for L can be approximated by $(\tilde{\beta}_0(\cdot), \tilde{\beta}(\cdot))$. Toward this end, let $\{\tilde{L}^{[k]}(\cdot, \cdot) : k \geq 1\}$ be a sequence of approximations to L such that

$$\lim_{k\to\infty} \sup_\eta \big|\tilde{L}_n^{[k]}(\eta) - L_n(\eta)\big| \to 0, \quad (10)$$

where

$$\tilde{L}_n^{[k]}(\eta) = \frac{1}{n}\sum_{i=1}^{n} \tilde{L}^{[k]}(y_i, \eta(\mathbf{x}_i)). \tag{11}$$

Clearly, (10) can be ensured if

$$\lim_{k\to\infty} \sup_{1\le i\le n,\eta} \bigl|\tilde{L}^{[k]}(y_i, \eta) - L(y_i, \eta)\bigr| \to 0. \tag{12}$$

Write

$$\bigl(\tilde{\beta}_0^{[k]}(\lambda), \tilde{\beta}^{[k]}(\lambda)\bigr) = \arg\min\left[\tilde{L}_n^{[k]}(\eta) + \lambda\sum_{j=1}^{p}|\beta_j|\right]. \tag{13}$$

We then have the following theorem concerning the approximation error of $(\tilde{\beta}_0^{[k]}(\lambda), \tilde{\beta}^{[k]}(\lambda))$.

Theorem 1. For any $\lambda \ge 0$ such that $(\beta_0(\lambda), \beta(\lambda))$ and $\{(\tilde{\beta}_0^{[k]}(\lambda), \tilde{\beta}^{[k]}(\lambda)) : k \ge 1\}$ are uniquely defined, we have

$$\tilde{\beta}_0^{[k]}(\lambda) \to \beta_0(\lambda), \qquad \tilde{\beta}^{[k]}(\lambda) \to \beta(\lambda) \tag{14}$$

as $k \to \infty$. Furthermore, if L_n is strictly convex in a neighborhood $\mathcal{N}$ around $(\beta_0(\lambda), \beta(\lambda))$, then, for any $\epsilon > 0$, there exist constants $k_0, C > 0$ such that for any $k \ge k_0$,

$$\bigl\|\tilde{\beta}^{[k]}(\lambda) - \beta(\lambda)\bigr\|_{\ell_2}^2 \le C\bigl|\tilde{L}_n^{[k]}(\eta) - L_n(\eta)\bigr|. \tag{15}$$

In Theorem 1, (14) qualitatively justifies the EGA path idea and (15) further quantifies its approximation accuracy. Now that we have approximated L by a quadratic spline $\tilde{L}$, it suffices to compute $(\tilde{\beta}_0(\cdot), \tilde{\beta}(\cdot))$. When using quadratic loss as in the multiple linear regression, Efron et al. (2004) developed the least-angle regression and showed that a simple modification of Lars yields the entire Lasso solution path. Their results can be extended to the ℓ_1-penalized quadratic spline loss. Following Efron et al. (2004), we first derive a generalized Lars algorithm, and then show that a simple modification of the generalized Lars algorithm can compute the entire solution path $(\tilde{\beta}_0(\cdot), \tilde{\beta}(\cdot))$.

2.2 Generalized Lars Algorithm

We start with no variables selected in the model. In this case, only the intercept is present; thus the starting point can be given as $\beta^{[1]} = \mathbf{0}$ and $\beta_0^{[1]} = \arg\min \tilde{L}_n(\beta_0)$. We next determine which variable should enter first. Without loss of generality, assume that $\kappa_{s_i}(y_i) < \beta_0^{[1]} < \kappa_{s_i+1}(y_i)$, $i = 1, 2, \ldots, n$. In a sufficiently small neighborhood around $(\beta_0^{[1]}, (\beta^{[1]})')'$,

$$\begin{aligned}
\tilde{L}_n(\beta_0, \beta) &\equiv \frac{1}{n}\sum_{i=1}^{n} \tilde{L}(y_i, \beta_0 + \mathbf{x}_i'\beta) \\
&= \frac{1}{n}\sum_{i=1}^{n}\bigl[a_{s_i}(y_i)\eta^2(\mathbf{x}_i) + b_{s_i}(y_i)\eta(\mathbf{x}_i) + c_{s_i}(y_i)\bigr] \\
&= \frac{1}{n}\sum_{i=1}^{n} a_{s_i}(y_i)\left(\beta_0 + \mathbf{x}_i'\beta + \frac{b_{s_i}(y_i)}{2a_{s_i}(y_i)}\right)^2 \\
&\quad + \frac{1}{n}\sum_{i=1}^{n}\left(c_{s_i}(y_i) - \frac{b_{s_i}(y_i)^2}{4a_{s_i}(y_i)}\right),
\end{aligned}$$

where

$$\begin{aligned}
a_j(Y) &= a_0(Y) + \sum_{j'\le j} d_{j'}(Y), \\
b_j(Y) &= b_0(Y) - 2\sum_{j'\le j} d_{j'}(Y)\kappa_{j'}(Y), \\
c_j(Y) &= c_0(Y) + \sum_{j'\le j} d_{j'}(Y)\kappa_{j'}^2(Y).
\end{aligned}$$

This amounts to a weighted linear regression of pseudoresponse,

$$\mathbf{y}^* = \left(-\frac{b_{s_1}(y_1)}{2a_{s_1}(y_1)}, -\frac{b_{s_2}(y_2)}{2a_{s_2}(y_2)}, \ldots, -\frac{b_{s_n}(y_n)}{2a_{s_n}(y_n)}\right)' \tag{16}$$

over the observed predictors $\mathbb{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)'$.

For n pairs of random variables $(z_{11}, z_{12}), \ldots, (z_{n1}, z_{n2})$ with case weights $w_1, \ldots, w_n \ge 0$, define the weighted correlation between them by

$$\operatorname{cov}^w(Z_1, Z_2) = \frac{1}{\sum_{i=1}^n w_i}\sum_{i=1}^{n} w_i(z_{i1} - \bar{z}_1^w)(z_{i2} - \bar{z}_2^w), \tag{17}$$

where

$$\bar{z}_j^w = \frac{\sum_{i=1}^n w_i z_{ij}}{\sum_{i=1}^n w_i} \tag{18}$$

is the weighted mean. In our case, set the weights to be $w_i = a_{s_i}(y_i)$, $i = 1, \ldots, n$. Then the predictor most weight-correlated with Y^* can reduce $\tilde{L}(\beta_0, \beta)$ most rapidly and should enter first. We define the active set $\mathcal{A} = \{j_1\}$, where X_{j_1} is the most weight-correlated with $\mathbf{y}^*$. We now move along the direction with only variable j_1 in the model. X_{j_1} remains the only variable that reduces $\tilde{L}_n(\beta_0, \beta)$ most rapidly until one of the following two possible events occurs:

(E1) Another variable j_2 has as much weighted correlation with "residual"

$$(y_1^* - \eta(\mathbf{x}_1), y_2^* - \eta(\mathbf{x}_2), \ldots, y_n^* - \eta(\mathbf{x}_n))' \tag{19}$$

as variable j_1.

(E2) An observation $(\mathbf{x}_i, y_i)$ leaves the s_ith segment, in that $\eta(\mathbf{x}_i) = \beta_0 + \mathbf{x}_i'\beta$ reaches either $\kappa_{s_i}(y_i)$ or $\kappa_{s_i+1}(y_i)$.

Figure 1 illustrates how the progression should be adjusted after these events occur. Similar to the original Lars, when (E1) occurs, we simply add j_2 to the active set $\mathcal{A}$, and the generalized Lars now proceeds in a direction with both X_{j_1} and X_{j_2} such that $\tilde{L}(\beta_0, \beta)$ reduces the fastest. This direction happens to be the direction equiangular between both predictors, as shown in Figure 1.

Now consider event (E2). Without loss of generality, assume that $\mathbf{x}_1$ reaches $\kappa_{s_1+1}(y_1)$. From this point on, $\eta(\mathbf{x}_1)$ falls into the $(s_1 + 1)$th segment of (7). Toward this end, we need to update s_1 by $s_1 + 1$. The change of segment also triggers the update of the pseudoresponse and weights; thus we need to adjust the pseudoresponse and each predictor accordingly so that the weighted mean remains 0,

$$y_i^* = y_i^* - (\bar{y}^*)^w, \qquad x_{ij}^* = x_{ij} - \bar{x}_j^w, \qquad j = 1, \ldots, p. \tag{20}$$

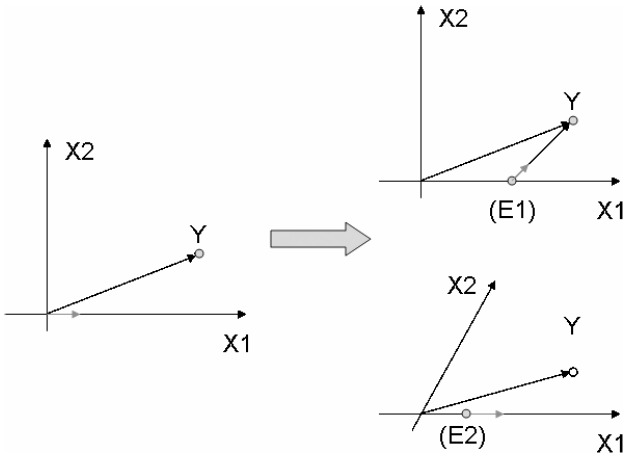

Figure 1. Adjustment when (E1) or (E2) occurs. As the generalized Lars algorithm proceeds, two types of events may occur. In (E1), X_2 has as much correlation with the residual as X_1. In (E2), one observation reaches the end of a segment and triggers the change of weight, which in turn results in the change of angle between X_1 and X_2.

It is not difficult to check that $X^*_{j_1}$ remains most correlated with the residual with the updated weights and responses. We continue marching in this direction. Basically, we recalibrate the angles between variables in response to (E2), as illustrated in Figure 1. The generalized Lars stops when $\tilde{L}_n(\beta_0, \beta)$ is minimized.

To summarize, the generalized Lars is given by the following algorithm:

Algorithm (Generalized Lars).

Step 1. Set $q = 1$ and $\beta^{[q]} = \mathbf{0}$. Compute $\beta_0^{[q]}$ as the minimizer of $\tilde{L}_n(\beta_0)$. Initialize $s_i^{[q]}$ so that $\kappa_{s_i^{[q]}} < \beta_0^{[q]} < \kappa_{s_i^{[q]}+1}$ for $i = 1, \ldots, n$.

Step 2. Update the following:

(a) Pseudoresponse:

$$\mathbf{y}^* = \left(-\frac{b_{s_1}(y_1)}{2a_{s_1}(y_1)}, -\frac{b_{s_2}(y_2)}{2a_{s_2}(y_2)}, \ldots, -\frac{b_{s_n}(y_n)}{2a_{s_n}(y_n)} \right)'. \tag{21}$$

(b) Weights:

$$w_i = a_{s_i}(y_i), \qquad i = 1, \ldots, n. \tag{22}$$

(c) Centered pseudoresponse and predictors:

$$\begin{aligned} y_i^* &= y_i^* - (\bar{y}^*)^w, \\ x_{ij}^* &= x_{ij} - \bar{x}_j^w, \qquad j = 1, \ldots, p. \end{aligned} \tag{23}$$

(d) Residual:

$$\mathbf{r}^{[q]} = \left(y_1^* - \mathbf{x}_1^* \beta^{[q]}, y_2^* - \mathbf{x}_2^* \beta^{[q]}, \ldots, y_n^* - \mathbf{x}_n^* \beta^{[q]} \right). \tag{24}$$

If $q = 1$, then actively set

$$\mathcal{A}^{[q]} = \arg\max_j \left\{ \left| \mathrm{cov}^w\left(X_j, \mathbf{r}^{[q]}\right) \right| \right\}. \tag{25}$$

Step 3. Compute the current direction, γ, which is a p-dimensional vector with $\gamma_{(\mathcal{A}^{[q]})^c} = \mathbf{0}$ and $\gamma_{\mathcal{A}^{[q]}}$ as the weighted least squares estimate when regressing $\mathbf{r}^{[q]}$ over $\mathbb{X}^*_{\mathcal{A}^{[q]}}$, where $\mathbb{X}^* = (x_{ij}^*)$ and the subscript $\mathcal{A}^{[q]}$ indicates the corresponding elements of a vector, or columns of a matrix are extracted. More specifically,

$$\gamma_{\mathcal{A}^{[q]}} = \left[\left(\mathbb{X}^*_{\mathcal{A}^{[q]}} \right)' \mathbf{W} \mathbb{X}^*_{\mathcal{A}^{[q]}} \right]^{-1} \left(\mathbb{X}^*_{\mathcal{A}^{[q]}} \right)' \mathbf{W} \mathbf{r}^{[q]}, \tag{26}$$

where $\mathbf{W} = \mathrm{diag}(w_1, \ldots, w_n)$.

Step 4. Compute how far the algorithm proceeds in the direction of γ until one of the following events occurs:

(a) A new variable j enters the active set. The distance α_j solves

$$\begin{aligned} &\left| (\mathbb{X}_j^*)' \mathbf{W} \mathbf{r}^{[q]} - \alpha_j (\mathbb{X}_j^*)' \mathbf{W} \mathbb{X}^* \gamma \right| \\ &\qquad = \left| (\mathbb{X}_a^*)' \mathbf{W} \mathbf{r}^{[q]} - \alpha_j (\mathbb{X}_a^*)' \mathbf{W} \mathbb{X}^* \gamma \right|, \end{aligned} \tag{27}$$

where $\mathbb{X}_j^*$ is the jth column of $\mathbb{X}^*$ and a is an arbitrary index in $\mathcal{A}^{[q]}$. It is not difficult to show that α_j can be given explicitly as

$$\alpha_j = \min_+ \left\{ \frac{(\mathbb{X}_j^* - \mathbb{X}_a^*)' \mathbf{W} \mathbf{r}^{[q]}}{(\mathbb{X}_j^* - \mathbb{X}_a^*)' \mathbf{W} \mathbb{X}^* \gamma}, \frac{(\mathbb{X}_j^* + \mathbb{X}_a^*)' \mathbf{W} \mathbf{r}^{[q]}}{(\mathbb{X}_j^* + \mathbb{X}_a^*)' \mathbf{W} \mathbb{X}^* \gamma} \right\}, \tag{28}$$

where $\min_+$ takes the minimum of the positive elements.

(b) $\eta(\mathbf{x}_i)$ leaves the $s_i^{[q]}$th segment of (7). Note that after moving $u_i \gamma$ from $\beta^{[q]}$, the intercept becomes

$$(\bar{y}^*)^w - (\bar{x}_1^w, \ldots, \bar{x}_n^w)\left(\beta^{[q]} + u_i \gamma \right). \tag{29}$$

Therefore, $\eta(\mathbf{x}_i)$ becomes

$$(\bar{y}^*)^w + (\mathbf{x}_i^*)'\left(\beta^{[q]} + u_i \gamma \right). \tag{30}$$

When $(\mathbf{x}_i^*)' \gamma > 0$, $\eta(\mathbf{x}_i)$ will reach $\kappa_{s_i+1}(y_i)$ with

$$u_i = \frac{1}{(\mathbf{x}_i^*)' \gamma} \left(\kappa_{s_i+1}(y_i) - (\bar{y}^*)^w - (\mathbf{x}_i^*)' \beta^{[q]} \right). \tag{31}$$

On the other hand, if $(\mathbf{x}_i^*)' \gamma < 0$, then $\eta(\mathbf{x}_i)$ will reach $\kappa_{s_i-1}(y_i)$ with

$$u_i = \frac{1}{(\mathbf{x}_i^*)' \gamma} \left(\kappa_{s_i-1}(y_i) - (\bar{y}^*)^w - (\mathbf{x}_i^*)' \beta^{[q]} \right). \tag{32}$$

Step 5. If a variable j enters the active set, then let $\alpha = \alpha_j$, and update $\mathcal{A}^{[q+1]} = \mathcal{A}^{[q]} \cup \{j\}$. If $\eta(\mathbf{x}_i)$ switches regions, then let $\alpha = u_i$, and update $s_i^{[q+1]} = s_i^{[q]} \pm 1$ accordingly.

Step 6. Update $\beta^{[q+1]} = \beta^{[q]} + \alpha \gamma$ and $q = q + 1$. Go back to step 2 if $\alpha < 1$.

When quadratic loss is used as in the multiple linear regression, (E2) will never occur, and the foregoing algorithm reduces to the original Lars.

2.3 ℓ_1-Regularized Quadratic Spline Loss

To take advantage of the generalized Lars algorithm, we first note that the ℓ_1-regularized solution path corresponding to $\tilde{L}$ is piecewise linear.

Theorem 2. Assume that $a_j \geq 0$ for all j. Then $(\tilde{\beta}_0(\cdot), \tilde{\beta}(\cdot))$ is piecewise linear.

Rosset and Zhu (2007) showed that the ℓ_1-penalized "almost" quadratic loss has piecewise linear solution paths. The loss function that they considered was piecewise quadratic in terms of the residual $(Y - \eta)$ in regression and the margin $Y\eta$ in classification. Theorem 2 is more general, in that it does not rely on the notion of residual or margin. Such generality becomes useful when we consider, for example, spline approximations to the negative log-likelihood for generalized linear models or the partial likelihood for Cox proportional hazards models.

Using the piecewise linear property of the solution paths corresponding to $\tilde{L}$, the solution paths can be computed by modifying the generalized Lars algorithm. The only difference occurs in steps 4 and 5, where another possible event, elimination of a selected variable, may occur. When this happens, we simply remove the variable and continue in the direction with the remaining variables in the active set. More specifically, we replace steps 4 and 5 in the generalized Lars by the following:

Algorithm (Generalized Lasso).

Step 4′. Compute how far the algorithm proceeds in the direction of γ until one of the following events occurs:

(a) A new variable j enters the active set. Define α_j as before.
(b) $\eta(\mathbf{x}_i)$ leaves the $s_i^{[q]}$th segment of (7). Define u_i as before.
(c) A variable j vanishes from $\mathcal{A}^{[q]}$. This occurs only when $\beta_j^{[q]}$ and γ_j are of opposite signs. Define $\alpha_j = -\beta_j^{[q]}/\gamma_j$.

Step 5′. If a variable j enters the active set, then denote $\alpha = \alpha_j$ and update $\mathcal{A}^{[q+1]} = \mathcal{A}^{[q]} \cup \{j\}$. If $\eta(\mathbf{x}_i)$ switches regions, then denote $\alpha = u_i$ and update $s_i^{[q+1]} = s_i^{[q]} \pm 1$ accordingly. If a variable j leaves the active set, then denote $\alpha = \alpha_j$ and update $\mathcal{A}^{[q+1]} = \mathcal{A}^{[q]} - \{j\}$.

The following theorem justifies our proposed algorithm.

Theorem 3. Under the "one at a time" condition described next, the foregoing algorithm produces the whole solution path $(\tilde{\beta}_0(\cdot), \tilde{\beta}(\cdot))$.

The term "one-at-a-time condition" was first used by Efron et al. (2004) in deriving the connection between Lasso and Lars for multiple linear regression. Similar conditions have been identified by Osborne, Presnell, and Turlach (2000). In our case, this term means that the events described in step 4′ do not occur simultaneously. This is generally true in practice and can always be enforced by slightly perturbing the response. (For more detailed discussions on this, see Osborne, Presnell, and Turlach 2000 or Efron et al. 2004.)

In principle, linear splines also could be used to approximate the loss function, in which case the resulting approximate solution paths would be piecewise linear as well, and a new generalized Lars algorithm could be devised for computing the linear spline loss solution paths. (Interested readers are referred to Yao and Lee 2007 for a recent investigation into problems of this type.) In the present work we prefer quadratic splines, for a couple of reasons. First and foremost, quadratic splines approximate the original loss function better than linear splines. For example, when $\partial^2 L/\partial \eta^2$ is Lipschitz of order α, linear splines with M knots generally have approximation errors of the order $M^{-(1+\alpha)}$, as opposed to $M^{-(2+\alpha)}$ for quadratic splines. In other words, to achieve similar approximation accuracy, many more knots are needed for linear splines than for quadratic splines. Note that the complexity of the generalized Lars algorithm depends on how often event (E1) or (E2) occurs; the latter happens more often as the number of knots increases. Therefore, quadratic spline loss can have tremendous computational advantages over linear spline loss when used for our purposes. Second, linear spline losses are nondifferentiable at the knots. As a result, many of the fitted values $\tilde{\eta}(\mathbf{x}_i) = \tilde{\beta}_0 + \mathbf{x}_i'\tilde{\beta}$, $i = 1, \ldots, n$, will take values at the knots of $\tilde{L}$. Such behavior is well known in contexts such as support vector machines. Although this is a desirable feature in these situations, it is more peculiar in our setting, because such "sticky" points are artifacts of the approximate loss, not of the original loss.

2.4 Example: ℓ_1-Regularized Logistic Regression

To illustrate the proposed methodology, we now consider an application to the ℓ_1-regularized logistic regression. Logistic regression falls into the more general class of generalized linear models where the loss function is given by

$$L(Y, \eta(\mathbf{X})) = -Y\eta(\mathbf{X}) + b[\eta(\mathbf{X})]. \tag{33}$$

For logistic regression, $b(\eta) = \ln[1 + \exp(\eta)]$. Clearly, the loss function is not a quadratic spline. To approximate it with a quadratic spline, it suffices to approximate the univariate function $b(\cdot)$ by a quadratic spline. This is a classical approximation problem that has been well studied in the literature (DeVore and Lorentz 1993). Consider approximating $b(\eta)$ by

$$\tilde{b}(\eta) = a_0\eta^2 + b_0\eta + c_0 + \sum_{j=1}^{M} d_j(\eta - \kappa_j)_+^2. \tag{34}$$

The parameters (i.e., the coefficients a_0, b_0, c_0, and d_j and the knots κ_j) in a quadratic spline can be solved numerically, that is, as the solution to the nonlinear optimization problem

$$\min_{a_0, b_0, c_0, d_j, \kappa_j} \left(\max_k |b(\eta_k) - \tilde{b}(\eta_k)| \right) \tag{35}$$

subject to the constraint that $a_0 + \sum_{j=1}^{J} d_j \geq 0$ for all $J \leq M$ to ensure convexity. Here $\{\eta_k\}$ is a set of prespecified values for evaluating the largest approximation error, usually a fine grid defined over the domain of $b(\cdot)$. The absolute error in (35) also can be replaced by other criteria that measure the approximation quality of $\tilde{b}(\cdot)$. Our experience suggests that the largest absolute error works pretty well in this case.

Take the logistic regression as an example. For $M = 2$, the following approximation can be obtained using the aforementioned strategy with 100 equally spaced η_k's in $[-5, 5]$:

$$\tilde{b}(\eta) = \begin{cases} 0.038, & \text{if } \eta \leq -2.77, \\ 0.09\eta^2 + 0.5\eta + 0.73, & \text{if } |\eta| \leq 2.77, \\ \eta + 0.038, & \text{if } \eta \geq 2.77. \end{cases} \tag{36}$$

Figure 2(a) shows $\tilde{b}(\cdot)$ together with $b(\cdot)$. The approximation error also can be evaluated:

$$\sup_{\eta, Y} |L(Y, \eta) - \tilde{L}(Y, \eta)| = \sup_{\eta} |b(\eta) - \tilde{b}(\eta)| = 0.038. \tag{37}$$

This approximation can be improved with an increasing number of knots; for example, the approximation error is decreased by about 13%, to 0.033, when $M = 4$. It is noteworthy

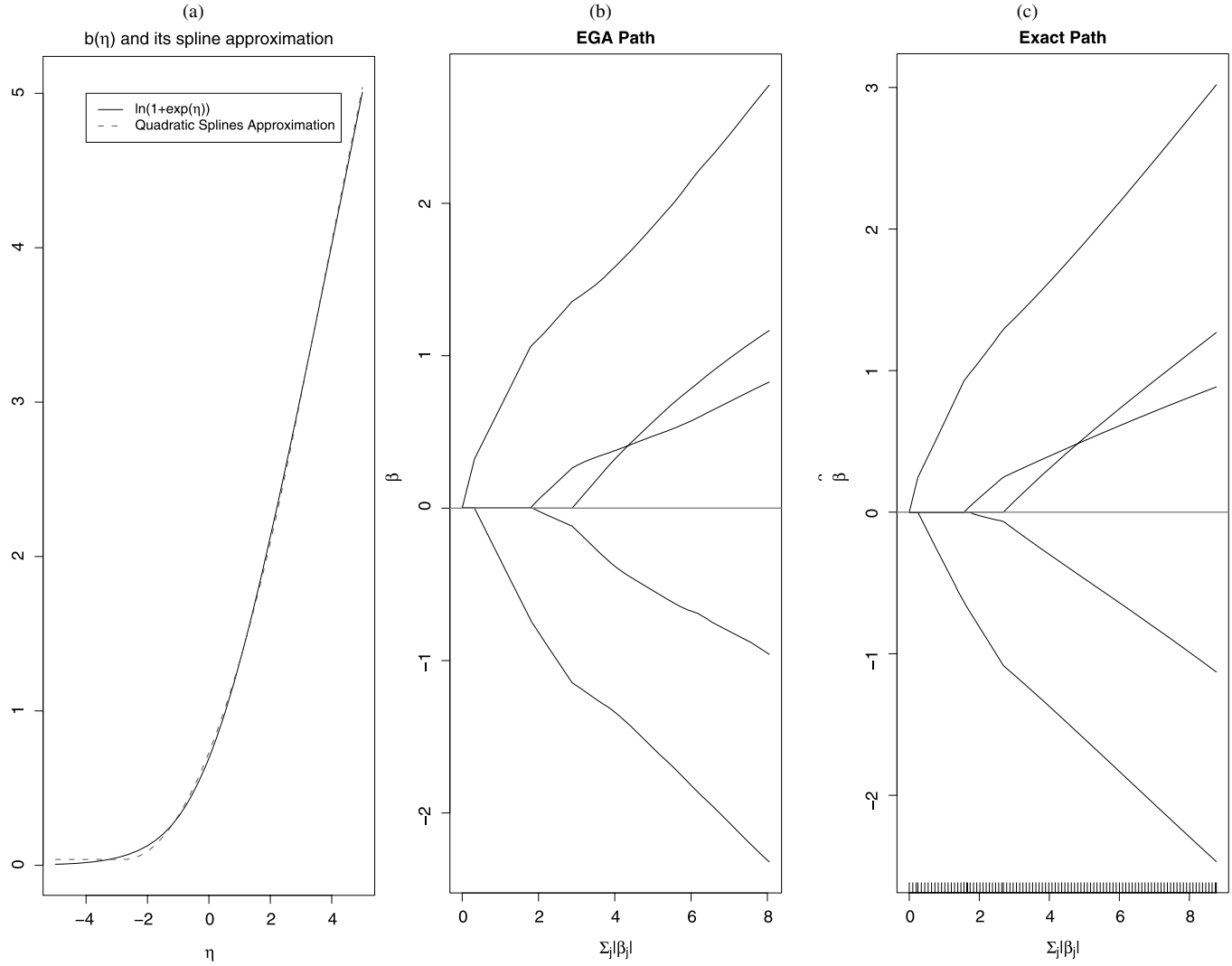

Figure 2. A side-by-side comparison of EGA path and exact (GLM) path in the synthetic example. (a) $b(\eta)$ together with its approximation (36). (b) The solution path constructed by EGA path. (c) The exact solution path computed by evaluating the ℓ_1-regularized estimate for the 200 regularization parameters marked by the ticks on the horizontal axis, and then linearly interpolating between the estimates.

that the larger the M, the more expensive the computation of $(\tilde{\beta}_0(\cdot), \tilde{\beta}(\cdot))$. In practice, a good choice of M reflects a trade-off between approximation accuracy and computational complexity. In the case of logistic loss, our experience suggests that $M = 2$ is a good choice; thus we use this approximation for the logistic regression loss throughout the rest of the article.

For illustration purposes, we consider a simulated example with $n = 50$ observations and $p = 5$ predictors. The predictors were generated from a multivariate normal distribution with mean 0 and $\text{cov}(X_i, X_j) = 0.5^{|i-j|}$. The true $\eta(\mathbf{X})$ is given by

$$\eta(X) = 2X_1 + X_2 - 2X_3 - X_4 + 0 \cdot X_5. \tag{38}$$

Figure 2 shows the EGA path and the exact solution path. The exact path was constructed by evaluating the exact ℓ_1-regularized estimate at 200 tuning parameters. The locations of the exact solutions are represented by the ticks on the horizontal axis. The figure clearly shows that the approximation works very well. We also ran the EGA path method with $M = 4$ knots and found no noticeable difference from the results reported here.

An alternative strategy for constructinf a piecewise linear approximation to the solution path of ℓ_1-regularized logistic regression was recently proposed by Park and Hastie (2007). The main idea of this so-called "GLM path" is to judiciously identify the "transition" points at which a variable is about to be added to or dropped from the model, and then use a nonlinear optimization solver to compute the exact solution at these transition points. The solution path is then constructed by linearly interpolating these exact solutions. The exact solution at each transition point is computed by Hessian-based iterative algorithms, which can be costly when p is large. In contrast, explicit formulas are available for each update of the proposed EGA path methodology, and an expensive nonlinear optimization solver is not needed. Besides reducing the computational cost, avoiding numerical optimization also provides stability in computing the EGA path. Hessian-based iterative algorithms for solving the ℓ_1-regularized logistic regression may be subject to numerical instability during intermediate iterations.

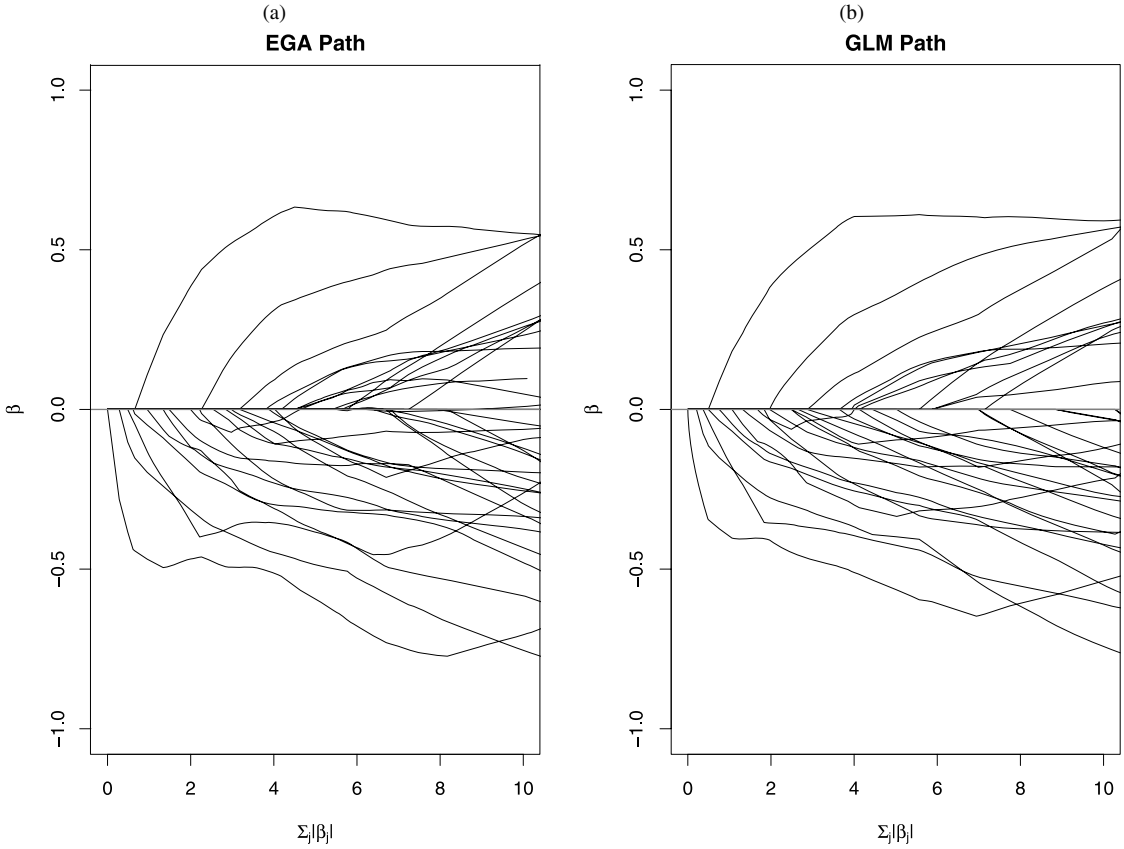

Figure 3. EGA path (a) and GLM path (b) for the Sonar data.

To demonstrate these differences, we applied both methods to the Sonar data, which were previously used by Gorman and Sejnowski (1988) in their study of the classification of sonar signals. The task is to train a classifier to discriminate between sonar signals bounced off a metal cylinder and those bounced off a roughly cylindrical rock based on a set of 60 numbers ranging from 0.0 to 1.0. Each number represents the energy within a particular frequency band, integrated over a certain time period. The integration aperture for higher frequencies occur later in time, because these frequencies are transmitted later during the chirp. There are a total of 208 signal observations, of which 97 were bounced off a rock and the remaining were bounced off a metal cylinder. We fit an ℓ_1-regularized logistic regression model to the Sonar data. The approximate solution paths are given in Figure 3. The similarity is evident; however, the computational advantage of the EGA path is noteworthy. An EGA path was constructed within 2.1 seconds, whereas constructing a GLM path with 200 regularization parameters took 6.3 seconds on the same computer. Also note that for 190 out of the 200 regularization parameters, the numerical optimization solver used by the GLM path encountered the problem of an ill-conditioned Hessian in intermediate iterations and gave convergence warnings.

3. OTHER ℓ_1–RELATED REGULARIZED PATHS

Our proposed methodology also can be easily extended to compute the global approximate solution paths of several ℓ_1-related regularization methods.

3.1 Generalized Elastic Net

The elastic net proposed by Zou and Hastie (2005) uses a mixed ℓ_1 and ℓ_2 regularization in the multiple linear regression. Zou and Hastie (2005) and subsequent studies (see, e.g., Park and Hastie 2007) demonstrated that the elastic net tends to yield more stable estimates than the usual ℓ_1-regularization in the presence of highly correlated predictors. Although the original elastic net was developed for multiple linear regression, the idea can be naturally extended to the more general predictive framework, where it can be formulated as

$$(\beta_0(\lambda_1,\lambda_2),\beta(\lambda_1,\lambda_2)) = \arg\min_{\beta_0,\beta}\left[\frac{1}{n}\sum_{i=1}^{n} L(y_i,\beta_0+\mathbf{x}_i'\beta) + \lambda_2\sum_{j=1}^{p}\beta_j^2 + \lambda_1\sum_{j=1}^{p}|\beta_j|\right]. \quad (39)$$

More generally, we can consider

$$(\beta_0(\lambda_1,\lambda_2),\beta(\lambda_1,\lambda_2)) = \arg\min_{\beta_0,\beta}\left[\frac{1}{n}\sum_{i=1}^{n} L(y_i,\beta_0+\mathbf{x}_i'\beta) + \lambda_2\beta'\Omega\beta + \lambda_1\sum_{j=1}^{p}|\beta_j|\right], \quad (40)$$

where Ω is a prespecified positive semidefinite $p \times p$ matrix. When Ω is the identity matrix, (40) reduces to the generalized elastic net. For a given λ_2, the solution path of the generalized

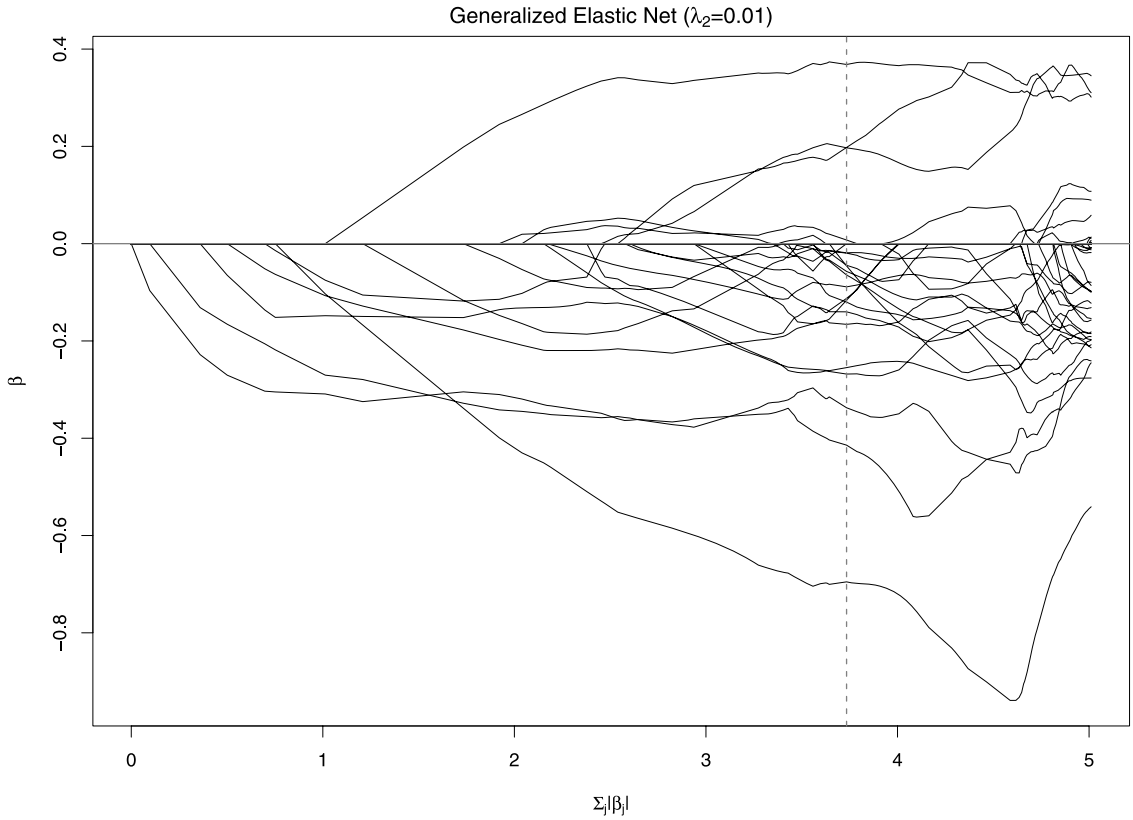

Figure 4. EGA path for the leukemia data. The gray vertical line corresponds to the tuning parameter chosen by 10-fold cross-validation.

elastic net, or, more generally (40), is nonlinear as a function of λ_1. To get an accurate piecewise linear approximation of the nonlinear solution path, we apply the EGA path idea by considering

$$(\tilde{\beta}_0(\lambda_1,\lambda_2),\tilde{\beta}(\lambda_1,\lambda_2)) = \arg\min\left[\tilde{L}_n(\eta)+\lambda_2\beta'\Omega\beta+\lambda_1\sum_{j=1}^{p}|\beta_j|\right], \quad (41)$$

where $\tilde{L}$ is the quadratic spline approximation to L. Following Theorem 1, it can be shown that the closeness between $\tilde{L}$ and L implies the closeness between $(\tilde{\beta}_0(\lambda_1,\lambda_2),\tilde{\beta}(\lambda_1,\lambda_2))$ and $(\beta_0(\lambda_1,\lambda_2),\beta(\lambda_1,\lambda_2))$ (see Appendix A.4). In practice, λ_2 often is chosen from a small number of candidates, that is, $\{0, 0.01, 0.1, 1, 10, 100\}$ (see, e.g., Zou and Hastie 2005). The impact of λ_1 is more complex, and the ability to construct the solution path indexed by λ_1 for each value of λ_2 is very important. The generalized Lars algorithm also can be easily modified for this purpose, as detailed in Appendix A.5.

To illustrate, first consider an application of the generalized elastic net (39) to the leukemia data set (Golub et al. 1999). This data set comprises 7129 genes and 72 samples; 38 of the samples, including 27 type 1 leukemia (acute lymphoblastic leukemia) and 11 type 2 leukemia (acute myeloid leukemia), are used as the training set. The goal is to construct a diagnostic rule based on the expression levels of the 7219 genes to predict the type of leukemia present. The remaining 34 samples are used as test set. We applied logistic loss with elastic net penalty (39) to the training data. As before, we used the spline approximation with $M = 2$. We used 10-fold cross-validation to jointly select tuning parameters λ_1 and λ_2; in particular, the value chosen for λ_2 was 0.01. Figure 4 shows the solution path as a function of λ_1 that corresponds to $\lambda_2 = 0.01$. The value of λ_1 selected is represented by the vertical broken line. The corresponding classification rule uses 21 genes. The cross- validation error was 1/38, whereas the test error was 2/34. The performance here is comparable to that in a previous analysis reported by Zou and Hastie (2005) and Park and Hastie (2007).

3.2 Support Vector Pursuit

We now discuss an interesting example of (40) that can identify support vectors in nonlinear discriminant analysis. Kernel methods are often used instead of linear methods to capture possible nonlinearity in $\eta(\cdot)$. This is often done in the framework of reproducing kernel Hilbert spaces. Let $\mathcal{H}_K$ be a reproducing kernel Hilbert space with kernel $K(\cdot,\cdot)$. The following regularization is commonly used to estimate $\eta(\cdot) = \beta_0 + \eta_1(\cdot)$ with $\eta_1(\cdot) \in \mathcal{H}_K$:

$$\eta_\lambda(\cdot) = \arg\min_{\beta_0\in\mathbb{R},\eta_1\in\mathcal{H}_K}\left[L_n(\eta)+\lambda\|\eta_1\|^2_{\mathcal{H}_K}\right]. \quad (42)$$

Estimates of this form have been well studied in the literature (Wahba 1990). It is known that although the minimization is taken over a possibly infinite-dimensional space $\mathcal{H}_K$, the solution of (42) actually lies in a finite-dimensional space and can be given by

$$\eta_\lambda(\cdot) = \beta_0(\lambda) + \sum_{i=1}^{n}\beta_i(\lambda)K(\cdot,\mathbf{x}_i), \quad (43)$$

and the β's can be obtained by putting this expression back into (42),

$$(\beta_0(\lambda), \beta(\lambda)) = \arg\min_{\beta} \left[\frac{1}{n} \sum_{i=1}^{n} L\left(y_i, \beta_0 + \sum_{j=1}^{n} \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \right) + \lambda \beta' K \beta \right], \quad (44)$$

where, with a slight abuse of notation, K denotes the Gram matrix with the (i, j) entry $K(\mathbf{x}_i, \mathbf{x}_j)$. For some loss functions, such as the hinge loss underlying the support vector machine, the solution in (44) has a sparse representation; that is, many $\hat{\beta}_j$'s are 0. Those observations with $\hat{\beta}_j \neq 0$ are often referred to as "support vectors." Other interesting loss functions do not enjoy such sparsity, however. A well-known example is kernel logistic regression. Various multistep procedures have been pursued to encourage sparsity in kernel logistic regression (Lin et al. 2000; Williams and Seeger 2001; Zhu and Hastie 2002; Zhang et al. 2004). A common idea is to find a sparse matrix approximation of the Gram matrix, and then fit the kernel logistic regression model using the sparse submatrix. Because an exclusive search is infeasible, some type of greedy algorithm is used to to search for a good sparse matrix approximation. This method is similar in spirit to forward/backward regression for multiple linear regression.

We show here that the idea of ℓ_1-regularization can be more naturally adopted to induce support vectors in kernel logistic regression. We term this technique *support vector pursuit*. In particular, we add an additional ℓ_1 penalty in (44),

$$(\beta_0(\lambda), \beta(\lambda)) = \arg\min_{\beta, \beta_0} \left[\frac{1}{n} \sum_{i=1}^{n} L\left(y_i, \beta_0 + \sum_{j=1}^{n} \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \right) + \lambda_2 \beta' K \beta + \lambda_1 \sum_{j=1}^{n} |\beta_j| \right], \quad (45)$$

where L is the logistic loss and λ_2 and λ_1 are two regularization parameters. It is worth mentioning that, to the best of our knowledge, no existing methods can be used to efficiently compute the support vector pursuit. Similar to the elastic net, we consider a small set of candidate values for λ_2; for example, let λ^* be the tuning parameter selected for (42), which often is done by 10-fold cross-validation or generalized cross-validation. We then consider the following candidate values for λ_2: $\{\lambda^*/100, \lambda^*/10, \lambda^*, 10\lambda^*, 100\lambda^*\}$. For each λ_2, we construct the EGA path for $0 \leq \lambda_1 < \infty$. The number of support vectors is controlled by λ_1; generally speaking, the larger λ_1, the fewer support vectors are used. Thus, compared with the import vector machine idea of Zhu and Hastie (2002), support vector pursuit generates support vectors in a smoother fashion.

To demonstrate the support vector pursuit method, we first consider a synthetic example with univariate smoothing spline kernel logistic regression. Take x_i, $i = 1, \ldots, 101$, to be equally spaced between 0 and 1. The responses y_i were generated from a binomial trial with probability of success $p(x_i) = 1 - 1/(1 + \exp(\eta(x_i)))$, where

$$\eta(X) = 3 \sin(2\pi X). \quad (46)$$

Following Wahba (1990), we let $\mathcal{H}_K$ be the orthogonal complement of constant functions in the second-order Sobolev–Hilbert space and choose the kernel so that

$$\|\eta_1\|^2_{\mathcal{H}_K} = \int (\eta_1'')^2 + \left(\int \eta_1' \right)^2. \quad (47)$$

For comparison, we ran both the usual kernel estimate given by (42) and the proposed support vector pursuit estimate on the simulated data. Figure 5 shows the true probability function $p(\cdot)$ and its estimates. All tuning parameters were chosen by 10-fold cross-validation. From Figure 5, we can see that the two estimates are essentially the same in terms of fitting the data. But all coefficients of the usual kernel estimate are nonzero, whereas the support vector pursuit relies on only eight support vectors, represented by the red circles.

We next applied the support vector pursuit to the mixture data example considered by Zhu and Hastie (2002) and Hastie, Tibshirani, and Friedman (2001). The training data were 200 data points generated from a pair of mixture densities. Because the Bayes classification boundary was nonlinear, we fitted kernel classifiers to the training data. Hastie et al. (2001) suggested using the radial kernel $K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\|\mathbf{x}_1 - \mathbf{x}_2\|^2_{\ell_2})$ to fit the support vector machine and kernel logistic regression. The support vector machine with the smallest test error (i.e., 0.218) had 107 support vectors. The import vector machine of Zhu and Hastie (2002) used 21 import vectors and had a test error was 0.219. Using the same kernel function, the support vector pursuit used 27 support vectors to achieve the test error of 0.213. Figure 6 shows the support vector pursuit decision boundary and the support vectors.

4. DISCUSSIONS

In this article we have proposed a global approach to approximating the nonlinear ℓ_1-regularization paths. We also have illustrated how the so-called "EGA path" idea can be applied to solve other interesting statistical learning problems, such as the generalized elastic net and support vector pursuit. The proposed methodology provides piecewise linear approximations to the true nonlinear regularization path. The EGA path's computational efficiency path facilitates the choice of the tuning parameter. If necessary, the EGA estimate also can be used as the initial value in an iterative nonlinear optimization solver to compute the exact solution.

While we were revising this manuscript, it was brought to our attention that Friedman, Hastie, and Tibshirani (2008) have developed an efficient `R` package, `glmnet`, for computing the ℓ_1-regularization path for logistic regression with the elastic-net penalty (with $\Omega = I$). `glmnet` can efficiently compute the solution at a given regularization parameter, and thus the whole process is repeated for typically 100 different regularization parameters to construct a piecewise linear approximation of the true nonlinear solution path. The performance of `glmnet` hinges on the choice of these regularization parameters, which clearly differs from those in `glmpath` and the EGA path. We also ran some experiments to compare the three methods on the Sonar data and Golub's data; the results are given in Table 1. Keep in mind that such a comparison may not faithfully represent the computational complexity of these algorithms due to

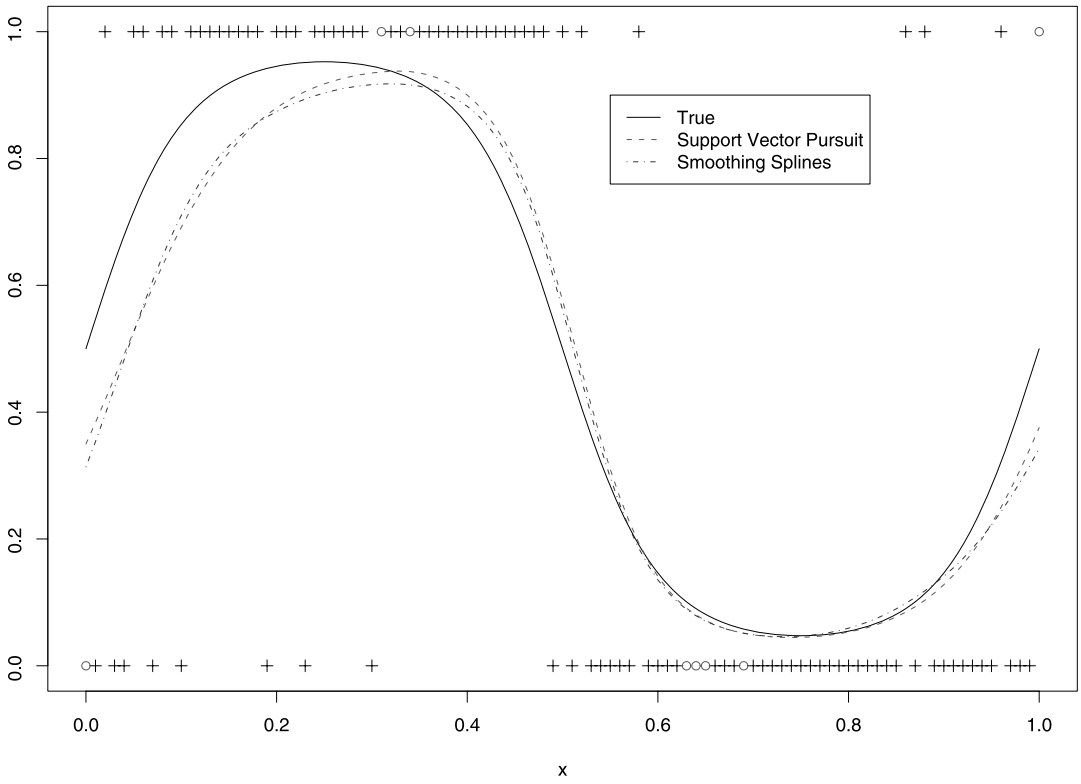

Figure 5. The smoothing spline kernel example for support vector pursuit. The eight circles correspond to the "support vectors" chosen by the support vector pursuit.

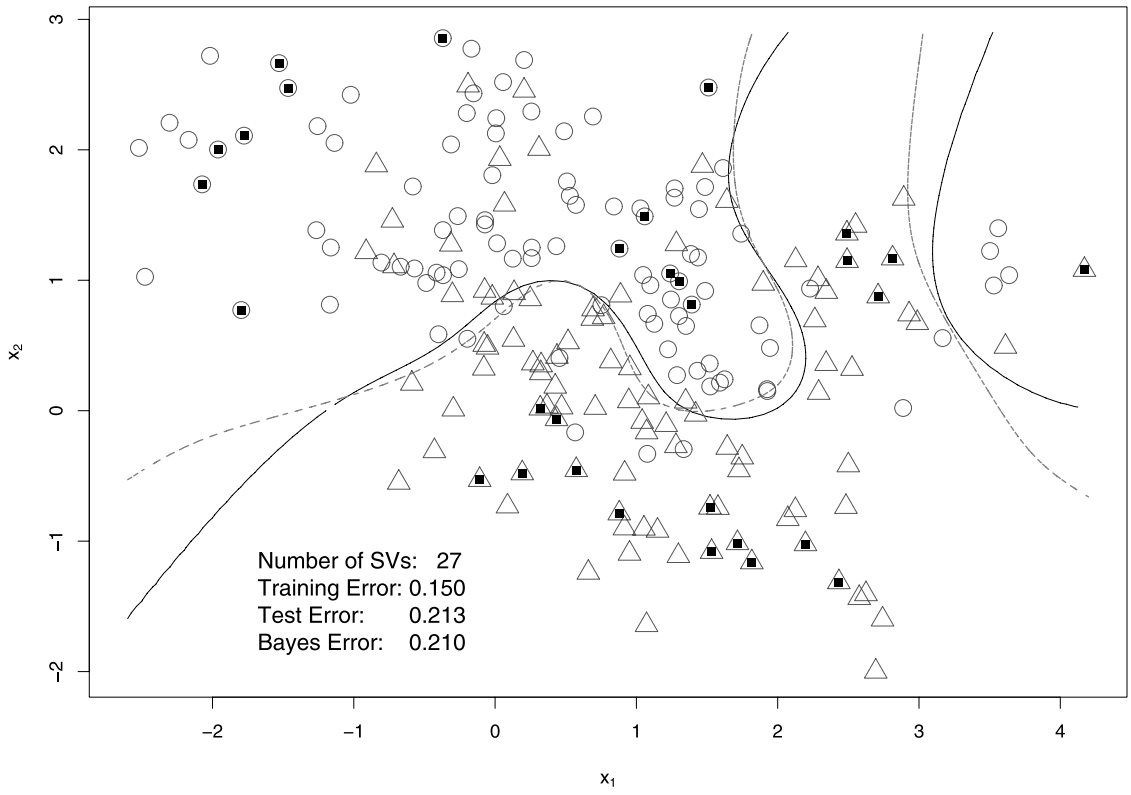

Figure 6. The mixture data example for the support vector pursuit. The training data in two classes are indicated by circles and triangles, respectively. The broken lines are the Bayes decision boundary. The dark solid lines are the support vector pursuit decision boundary. Solid black squares are the 27 support vectors.

Table 1. Times (in seconds) for the `glmnet`, `glmpath`, and EGA paths

	Sonar	Golub
`glmnet`	10.59	6.08
`glmpath`	6.33	14.45
EGA	2.09	9.05

NOTE: All timings were carried out on an Intel Core 2 Duo Processor E8200 2.66 GHz.

implementation differences. EGA does all of its numeric computations in `R`, `glmnet` does all of its computations in `Fortran`, and `glmpath` frequently calls `Fortran` routines to do numerical optimization. Nevertheless, Table 1 indicates that these methods are of comparable computational speed for these two data sets.

Finally, we want to point out that the EGA path idea also can be easily extended to handle other challenging nonlinear regularization problems using concave penalties. For example, the SCAD penalty for model selection and estimation of Fan and Li (2001, 2006) is often solved by iterative algorithms for a grid of regularization parameters. Recently, Zou and Li (2008) proposed the LLA algorithm for solving the SCAD. Combining the LLA algorithm and our EGA path can boost the computational efficiency of the SCAD estimator in, for example, the SCAD-penalized logistic regression model.

APPENDIX: PROOFS OF THEOREMS AND APPROXIMATION ERROR AND ALGORITHM OF THE EGA PATH FOR THE GENERALIZED ELASTIC NET

A.1 Proof of Theorem 1

For brevity, we write β instead of (β_0, β) when no confusion occurs. Let $\beta^*(\lambda) \neq \beta(\lambda)$ be the limit of an arbitrary convergent subsequence $\{\tilde\beta^{[k_l]}(\lambda)\}$ of $\{\tilde\beta^{[k]}(\lambda)\}$. By definition,

$$\tilde L_n^{[k_l]}\bigl(\tilde\beta^{[k_l]}(\lambda)\bigr) + \lambda\sum_{j=1}^{p}\bigl|\tilde\beta_j^{[k_l]}(\lambda)\bigr| \le \tilde L_n^{[k_l]}(\beta(\lambda)) + \lambda\sum_{j=1}^{p}|\beta_j(\lambda)|. \quad (A.1)$$

Taking the limit of both sides yields

$$L_n(\beta^*(\lambda)) + \lambda\sum_{j=1}^{p}|\beta_j^*(\lambda)| \le L_n(\beta(\lambda)) + \lambda\sum_{j=1}^{p}|\beta_j(\lambda)|, \quad (A.2)$$

which implies that $\beta^*(\lambda) = \beta(\lambda)$ by the definition of $\beta(\lambda)$. Therefore, $\tilde\beta^{[k]}(\lambda) \to \beta(\lambda)$ as k goes to infinity. Thus when k is sufficiently large, $\tilde\beta^{[k]}(\lambda)$ falls into the neighborhood of $\beta(\lambda)$, $\mathcal{N}$.

Denote $\mathcal{A} = \{j : \beta_j(\lambda) \neq 0\}$. By Taylor's expansion,

$$\begin{aligned}
&\left(L_n\bigl(\tilde\beta^{[k]}(\lambda)\bigr) + \lambda\sum_{j=1}^{p}\bigl|\tilde\beta_j^{[k]}(\lambda)\bigr|\right) - \left(L_n(\beta(\lambda)) + \lambda\sum_{j=1}^{p}|\beta_j(\lambda)|\right)\\
&= \sum_{j\notin\mathcal{A}}\left[\left.\frac{\partial L_n}{\partial\beta_j}\right|_{\beta(\lambda)}\tilde\beta_j^{[k]}(\lambda) + \lambda\bigl|\tilde\beta_j^{[k]}(\lambda)\bigr|\right]\\
&\quad + \sum_{j\in\mathcal{A}}\left[\left.\frac{\partial L_n}{\partial\beta_j}\right|_{\beta(\lambda)}\bigl(\tilde\beta_j^{[k]}(\lambda) - \beta_j(\lambda)\bigr) + \lambda\bigl(\bigl|\tilde\beta_j^{[k]}(\lambda)\bigr| - |\beta_j(\lambda)|\bigr)\right]\\
&\quad + \frac{1}{2}\bigl(\tilde\beta^{[k]}(\lambda) - \beta(\lambda)\bigr)'\left.\frac{\partial^2 L_n}{\partial\beta\,\partial\beta'}\right|_{t\beta(\lambda)+(1-t)\tilde\beta^{[k]}(\lambda)}\\
&\quad \times\bigl(\tilde\beta^{[k]}(\lambda) - \beta(\lambda)\bigr)
\end{aligned}$$

for some $0 \le t \le 1$. First-order conditions indicate that

$$\left|\left.\frac{\partial L_n}{\partial\beta_j}\right|_{\beta(\lambda)}\right| \le \lambda \quad \text{if } j\notin\mathcal{A}, \quad (A.3)$$

$$\left.\frac{\partial L_n}{\partial\beta_j}\right|_{\beta(\lambda)} = \lambda\,\mathrm{sign}(\beta_j(\lambda)) \quad \text{if } j\in\mathcal{A}. \quad (A.4)$$

Equation (A.3) implies that for any $j\notin\mathcal{A}$,

$$\left.\frac{\partial L_n}{\partial\beta_j}\right|_{\beta(\lambda)}\tilde\beta_j^{[k]}(\lambda) + \lambda\bigl|\tilde\beta_j^{[k]}(\lambda)\bigr| \ge 0. \quad (A.5)$$

Because $\tilde\beta^{[k]}(\lambda) \to \beta(\lambda)$, there exists a k^* such that for $k > k^*$, $\mathrm{sign}(\tilde\beta_j^{[k]}(\lambda)) = \mathrm{sign}(\beta_j(\lambda))$ for all $j\in\mathcal{A}$. Together with (A.4),

$$\left.\frac{\partial L_n}{\partial\beta_j}\right|_{\beta(\lambda)}\bigl(\tilde\beta_j^{[k]}(\lambda) - \beta_j(\lambda)\bigr) + \lambda\bigl(\bigl|\tilde\beta_j^{[k]}(\lambda)\bigr| - |\beta_j(\lambda)|\bigr) = 0 \quad (A.6)$$

for any $j\in\mathcal{A}$ and $k > k^*$. Therefore,

$$\begin{aligned}
&\left(L_n\bigl(\tilde\beta^{[k]}(\lambda)\bigr) + \lambda\sum_{j=1}^{p}\bigl|\tilde\beta_j^{[k]}(\lambda)\bigr|\right) - \left(L_n(\beta(\lambda)) + \lambda\sum_{j=1}^{p}|\beta_j(\lambda)|\right)\\
&\qquad \ge C\bigl\|\tilde\beta^{[k]}(\lambda) - \beta(\lambda)\bigr\|_{\ell_2}^2 \quad (A.7)
\end{aligned}$$

for some constant $C > 0$, because $\partial^2 L_n/\partial\beta\,\partial\beta'$ is strictly positive definite on $\mathcal{N}$.

In contrast, by definition,

$$\begin{aligned}
&\tilde L_n^{[k]}\bigl(\tilde\beta^{[k]}(\lambda)\bigr) + \lambda\sum_{j=1}^{p}\bigl|\tilde\beta_j^{[k]}(\lambda)\bigr|\\
&\le \tilde L_n^{[k]}(\beta(\lambda)) + \lambda\sum_{j=1}^{p}|\beta_j(\lambda)|\\
&= \bigl[\tilde L_n^{[k]}(\beta(\lambda)) - L_n(\beta(\lambda))\bigr] + \left[L_n(\beta(\lambda)) + \lambda\sum_{j=1}^{p}|\beta_j(\lambda)|\right],
\end{aligned}$$

which implies that

$$\begin{aligned}
&\left(L_n\bigl(\tilde\beta^{[k]}(\lambda)\bigr) + \lambda\sum_{j=1}^{p}\bigl|\tilde\beta_j^{[k]}(\lambda)\bigr|\right) - \left(L_n(\beta(\lambda)) + \lambda\sum_{j=1}^{p}|\beta_j(\lambda)|\right)\\
&\qquad \le \bigl[\tilde L_n^{[k]}(\beta(\lambda)) - L_n(\beta(\lambda))\bigr]. \quad (A.8)
\end{aligned}$$

This, together with (A.7), yields the desirable result.

A.2 Proof of Theorem 2

It is not hard to see that $\tilde L$ given by (7) is convex given that $a_j \ge 0$ for all j. Thus, from the Karush–Kuhn–Tucker theorem, $(\tilde\beta_0(\lambda), \tilde\beta(\lambda))$ satisfies

$$\frac{1}{n}\sum_{i=1}^{n}x_{ij}\bigl[2a_{s_i}(y_i)\tilde\eta(\mathbf{x}_i) + b_{s_i}\bigr] + \lambda\,\mathrm{sign}(\tilde\beta_j(\lambda)) = 0 \quad \text{if } \tilde\beta_j(\lambda)\neq 0,$$

$$\left|\frac{1}{n}\sum_{i=1}^{n}x_{ij}\bigl[2a_{s_i}(y_i)\tilde\eta(\mathbf{x}_i) + b_{s_i}\bigr]\right| \le \lambda \quad \text{if } \tilde\beta_j(\lambda) = 0,$$

where $\tilde\eta(\mathbf{x}_i) = \tilde\beta_0 + \mathbf{x}_i'\tilde\beta$. Hereinafter we suppress the dependence of $\tilde\beta$ on λ if no confusion occurs. Write $\mathcal{A} = \{j : \tilde\beta_j \neq 0\}$, and also write $\mathbb{X}_{\mathcal{B}} = (\mathbf{1}, \mathbb{X}_j : j\in\mathcal{A})$, $s_{\mathcal{B}} = (0, \mathrm{sign}(\tilde\beta_j) : j\in\mathcal{A})'$ and $\mathbf{W} = \mathrm{diag}(a_{s_i}(y_i) : i = 1,\ldots,n)$. Then, from the foregoing Karush–Kuhn–Tucker conditions, it can be deduced that $\tilde\beta_{\mathcal{A}^c} = \mathbf{0}$ and

$$\begin{aligned}
\tilde\beta_{\mathcal{B}} &\equiv (\tilde\beta_0, \tilde\beta_j : j\in\mathcal{A})'\\
&= (\mathbb{X}_{\mathcal{B}}'\mathbf{W}\mathbb{X}_{\mathcal{B}})^{-1}\mathbb{X}_{\mathcal{B}}'\mathbf{b} - \frac{n\lambda}{2}(\mathbb{X}_{\mathcal{B}}'\mathbf{W}\mathbb{X}_{\mathcal{B}})^{-1}s_{\mathcal{B}}, \quad (A.9)
\end{aligned}$$

where $\mathbf{b} = (b_{s_1}, \ldots, b_{s_n})'$. It is now evident that the $\tilde{\beta}$ indexed by λ has a linear trajectory unless one of the following two events occurs:

(a) $\mathcal{A}$ changes, either with an existing variable leaving $\mathcal{A}$ or another new variable entering $\mathcal{A}$

(b) s_i changes for some $i - \tilde{\beta}_0 + \mathbf{x}_i\tilde{\beta}$ reaches either bounds of its corresponding segment.

The proof is completed.

A.3 Proof of Theorem 3

It suffices to show that any point on the constructed solution path satisfies the Karush–Kuhn–Tucker conditions for some $\lambda \geq 0$. This is clearly true for the starting point $\tilde{\beta} = \mathbf{0}$. In the light of Theorem 2, we need to show that this remains true after either $\mathcal{A}$ or any s_i changes. When $\mathcal{A}$ changes, the proof follows from the same argument as for the Lasso (Efron et al. 2004) and thus is omitted here. Now consider the case when s_i changes. Without loss of generality, assume that s_1 changes to $s_1 + 1$, in other words, $\tilde{\eta}(\mathbf{x}_1)$ increases to $\kappa_1(y_1)$. Thus it is necessary that

$$\mathbf{x}_{1\mathcal{B}}(\mathbb{X}'_{\mathcal{B}}\mathbf{W}\mathbb{X}_{\mathcal{B}})^{-1}s_{\mathcal{B}} > 0 \tag{A.10}$$

for $\tilde{\eta}(\mathbf{x}_1)$ to increase. For the algorithm to work appropriately, we need to show that the the first observation will not come back to the s_1th segment immediately. To show this, assume the contrary, that for some tuning parameter $\lambda - \epsilon$ with $\epsilon > 0$ arbitrarily small, $\tilde{\eta}^{[\lambda-\epsilon]}(\mathbf{x}_1) < \tilde{\eta}^{[\lambda]}(\mathbf{x}_1)$, where the superscripts represent the corresponding tuning parameters. With ϵ sufficiently small, we can assume that the $\mathcal{B}$'s remain unchanged; therefore,

$$\begin{aligned}
\tilde{\eta}^{[\lambda-\epsilon]}(\mathbf{x}_1) &= \mathbf{x}_{1\mathcal{B}}\tilde{\beta}_{\mathcal{B}}(\lambda+\epsilon) \\
&= \mathbf{x}'_{1\mathcal{B}}(\mathbb{X}'_{\mathcal{B}}\mathbf{W}\mathbb{X}_{\mathcal{B}})^{-1}\mathbb{X}'_{\mathcal{B}}\mathbf{b} \\
&\quad - \frac{n(\lambda-\epsilon)}{2}\mathbf{x}'_{1\mathcal{B}}(\mathbb{X}'_{\mathcal{B}}\mathbf{W}\mathbb{X}_{\mathcal{B}})^{-1}s_{\mathcal{B}} \\
&\geq \mathbf{x}'_{1\mathcal{B}}(\mathbb{X}'_{\mathcal{B}}\mathbf{W}\mathbb{X}_{\mathcal{B}})^{-1}\mathbb{X}'_{\mathcal{B}}\mathbf{b} - \frac{n\lambda}{2}\mathbf{x}'_{1\mathcal{B}}(\mathbb{X}'_{\mathcal{B}}\mathbf{W}\mathbb{X}_{\mathcal{B}})^{-1}s_{\mathcal{B}} \\
&= \tilde{\eta}^{[\lambda]}(\mathbf{x}_1),
\end{aligned}$$

which contradicts the assumption stated earlier. Therefore, the first observation will not return to the s_1th segment immediately. This completes the proof of the theorem.

A.4 Approximation Error of the EGA Path for the Generalized Elastic Net

Similar to before, let $\{\tilde{L}^{[k]} : k \geq 1\}$ be a sequence of approximations to L such that

$$\lim_{k\to\infty} \sup_{\eta} \left|\tilde{L}_n^{[k]}(\eta) - L_n(\eta)\right| \to 0. \tag{A.11}$$

Denote by $(\tilde{\beta}_0^{[k]}(\lambda_1, \lambda_2), \tilde{\beta}^{[k]}(\lambda_1, \lambda_2))$ the solution corresponding to loss $\tilde{L}^{[k]}$. By Theorem 1 and Corollary 1, we have the following.

Corollary 1. For any $\lambda_1, \lambda_2 \geq 0$ such that $(\beta_0(\lambda_1, \lambda_2), \beta(\lambda_1, \lambda_2))$ and $\{(\tilde{\beta}_0^{[k]}(\lambda_1, \lambda_2), \tilde{\beta}^{[k]}(\lambda_1, \lambda_2)) : k \geq 1\}$ are uniquely defined, we have

$$\tilde{\beta}_0^{[k]}(\lambda_1, \lambda_2) \to \beta_0(\lambda_1, \lambda_2), \qquad \tilde{\beta}^{[k]}(\lambda_1, \lambda_2) \to \beta(\lambda_1, \lambda_2) \tag{A.12}$$

as $k \to \infty$. Furthermore, if L_n is strictly convex in a neighborhood $\mathcal{N}$ around $(\beta_0(\lambda_1, \lambda_2), \beta(\lambda_1, \lambda_2))$, then for any $\epsilon > 0$, there exist constants $k_0, C > 0$ such that for any $k \geq k_0$,

$$\begin{aligned}
\left\|\tilde{\beta}^{[k]}(\lambda_1, \lambda_2) - \beta(\lambda_1, \lambda_2)\right\|_{\ell_2}^2 \leq C\big|&\tilde{L}_n^{[k]}(\beta_0(\lambda_1, \lambda_2), \beta(\lambda_1, \lambda_2)) \\
&- L_n(\beta_0(\lambda_1, \lambda_2), \beta(\lambda_1, \lambda_2))\big|.
\end{aligned} \tag{A.13}$$

A.5 Algorithm of the EGA Path for the Generalized Elastic Net

To compute the solution path for (41), we need only to modify the evaluation of the current direction γ and distances α_j in the generalized Lasso algorithm presented in Section 2.3. Recall that

$$\begin{aligned}
\tilde{L}(\beta_0, \beta) = \frac{1}{n}\sum_{i=1}^{n} a_{s_i}(y_i)(y_i^* - (\beta_0 + \mathbf{x}_i'\beta))^2 \\
+ \frac{1}{n}\sum_{i=1}^{n}\left(c_{s_i}(y_i) - \frac{b_{s_i}(y_i)^2}{4a_{s_i}(y_i)}\right).
\end{aligned} \tag{A.14}$$

Without the additional penalty, the direction taken at the qth iteration, $\gamma_{\mathcal{A}^{[q]}}$, can be given as the minimizer of the weighted least squares,

$$\frac{1}{n}\sum_{i=1}^{n} a_{s_i}(y_i)(y_i^* - (\mathbf{x}^*_{i,\mathcal{A}^{[q]}})'(\beta_{\mathcal{A}^{[q]}} + \gamma))^2. \tag{A.15}$$

When the additional penalty is present, $\gamma_{\mathcal{A}^{[q]}}$ now becomes the minimizer of

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} a_{s_i}(y_i)(y_i^* - (\mathbf{x}^*_{i,\mathcal{A}^{[q]}})'(\beta_{\mathcal{A}^{[q]}} + \gamma_{\mathcal{A}^{[q]}}))^2 \\
+ \lambda_2(\beta_{\mathcal{A}^{[q]}} + \gamma_{\mathcal{A}^{[q]}})'\Omega(\beta_{\mathcal{A}^{[q]}} + \gamma_{\mathcal{A}^{[q]}}).
\end{aligned} \tag{A.16}$$

It is not hard to see that the minimizer is

$$\gamma_{\mathcal{A}^{[q]}} = [(\mathbb{X}^*_{\mathcal{A}^{[q]}})'\mathbf{W}\mathbb{X}^*_{\mathcal{A}^{[q]}} + \lambda\Omega]^{-1}(\mathbb{X}^*_{\mathcal{A}^{[q]}})'\mathbf{W}\mathbf{y}^* - \beta_{\mathcal{A}^{[q]}}. \tag{A.17}$$

Similarly, the calculation of the step length, α_j , also must be modified. An application of the Karush–Kuhn–Tucker conditions yields

$$\begin{aligned}
&\left|(\mathbb{X}_j^*)'\mathbf{W}\mathbf{r}^{[q]} - \alpha_j(\mathbb{X}_j^*)'\mathbf{W}\mathbb{X}^*\gamma - \lambda_2\big(\beta_j^{[q]} + \alpha_j\gamma_j\big)\right| \\
&\qquad = \left|(\mathbb{X}_a^*)'\mathbf{W}\mathbf{r}^{[q]} - \alpha_j(\mathbb{X}_a^*)'\mathbf{W}\mathbb{X}^*\gamma - \lambda_2\big(\beta_a^{[q]} + \alpha_j\gamma_a\big)\right|,
\end{aligned}$$

where, similar to the generalized Lars algorithm, a is an arbitrary index in $\mathcal{A}^{[q]}$. Denote $\delta_j = \beta_j^{[q]} + \gamma_j$ and $\delta_a = \beta_a^{[q]} + \gamma_a$. Then

$$\begin{aligned}
\alpha_j = \min_{+}\Bigg\{ &\frac{(\mathbb{X}_j^* - \mathbb{X}_a^*)'\mathbf{W}\mathbf{r}^{[q]} - \lambda_2(\beta_j^{[q]} - \beta_a^{[q]})}{(\mathbb{X}_j^* - \mathbb{X}_a^*)'\mathbf{W}\mathbb{X}^*\gamma + \lambda_2(\gamma_j - \gamma_a)}, \\
&\frac{(\mathbb{X}_j^* + \mathbb{X}_a^*)'\mathbf{W}\mathbf{r}^{[q]} - \lambda_2(\beta_j^{[q]} + \beta_a^{[q]})}{(\mathbb{X}_j^* + \mathbb{X}_a^*)'\mathbf{W}\mathbb{X}^*\gamma + \lambda_2(\gamma_j + \gamma_a)}\Bigg\}.
\end{aligned} \tag{A.18}$$

[Received May 2008. Revised June 2009.]

REFERENCES

DeVore, R., and Lorentz, G. (1993), *Constructive Approximation*, New York: Springer.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), "Least Angle Regression," *The Annals of Statistics*, 32, 407–499.

Fan, J., and Li, R. (2001), "Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties," *Journal of the American Statistical Association*, 96, 1348–1360.

——— (2006), "Statistical Challenges With High Dimensionality: Feature Selection in Knowledge Discovery," in *Proceedings of the International Congress of Mathematicians*, Vol. III, eds. M. Sanz-Sole, J. Soria, J. L. Varona, and J. Verdera, Zurich: European Mathematical Society, pp. 595–622.

Friedman, J., Hastie, T., and Tibshirani, R. (2008), "Regularization Paths for Generalized Linear Models via Coordinate Descent," technical report, Stanford University, Dept. of Statistics.

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.

Gorman, R. P., and Sejnowski, T. J. (1988), "Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets," *Neural Networks*, 1, 75–89.

Hastie, T., Rosset, S., Tibshirani, R., and Zhu, J. (2005), "The Entire Regularization Path for the Support Vector Machine," *Journal of Machine Learning Research*, 5, 1391–1415.

Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, New York: Springer-Verlag.

Li, Y., and Zhu, J. (2008), "L_1-Norm Quantile Regression," *Journal of Computational and Graphical Statistics*, 17, 163–185.

Lin, X., Wahba, G., Xiang, D., Gao, F., Klein, R., and Klein, B. (2000), "Smoothing Spline ANOVA Models for Large Data Sets With Bernoulli Observations and the Randomized GACV," *The Annals of Statistics*, 28, 1570–1600.

Osborne, M., Presnell, B., and Turlach, B. (2000), "A New Approach to Variable Selection in Least Squares Problems," *IMA Journal of Numerical Analysis*, 20, 389–403.

Park, M., and Hastie, T. (2007), "L_1 Regularization Path Algorithm for Generalized Linear Models," *Journal of the Royal Statistical Society, Ser. B*, 69, 659–677.

Rosset, S. (2005), "Following Curved Regularized Optimization Solution Paths," in *Advances in Neural Information Processing Systems*, Vol. 17, Cambridge, MA: MIT Press.

Rosset, S., and Zhu, J. (2007), "Piecewise Linear Regularized Solution Paths," *The Annals of Statistics*, 35, 1012–1030.

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society, Ser. B*, 58, 267–288.

Wahba, G. (1990), *Spline Models for Observational Data*, Philadelphia: SIAM.

Williams, C., and Seeger, M. (2001), "Using the Nyström Method to Speed up Kernel Machines," *Advances in Neural Information Processing Systems*, Vol. 13, Cambridge, MA: MIT Press.

Yao, Y., and Lee, Y. (2007), "Another Look at Linear Programming for Feature Selection via Methods of Regularization," Technical Report 800, The Ohio State University, Dept. of Statistics.

Zhang, H., Wahba, G., Lin, Y., Voelker, M., Ferris, M., Klein, R., and Klein, B. (2004), "Variable Selection and Model Building via Likelihood Basis Pursuit," *Journal of the American Statistical Association*, 99, 659–672.

Zhao, P., and Yu, B. (2007), "Stagewise Lasso," *Journal of Machine Learning Research*, 8, 2701–2726.

Zhu, J., and Hastie, T. (2002), "Kernel Logistic Regression and the Import Vector Machine," in *Advances in Neural Information Processing Systems*, Vol. 14, Cambridge, MA: MIT Press.

Zhu, J., Rosset, S., Hastie, T., and Tibshirani, R. (2003), "1-Norm Support Vector Machines," in *Advances in Neural Information Processing Systems*, Vol. 16, Cambridge, MA: MIT Press.

Zou, H., and Hastie, T. (2005), "Regression and Variable Election via the Elastic Net," *Journal of the Royal Statistical Society, Ser. B*, 67, 301–320.

Zou, H., and Li, R. (2008), "One-Step Sparse Estimates in Nonconcave Penalized Likelihood Models," *The Annals of Statistics*, 36, 1509–1533.