EEBoost: A general method for prediction and variable selection based on estimating equations

Julian Wolfson [julianw@umn.edu]

Division of Biostatistics

University of Minnesota School of Public Health

February 17, 2010

Abstract

The modern statistical literature is replete with methods for performing variable selection and prediction in standard regression problems. However, simple models may misspecify or fail to capture important aspects of the data generating process such as missingness, correlation, and over/underdispersion. This realization has motivated the development of a large class of estimating equations which account for these data characteristics and often yield improved inference for low-dimensional parameters. In this paper we introduce EEBoost, a novel strategy for variable selection and prediction which can be applied in any problem where inference would typically be based on estimating

equations. The method is simple, flexible, and easily implemented using existing software.

Extended abstract

The modern statistical literature is replete with methods for performing variable selection and prediction in standard regression problems. However, simple models may misspecify or fail to capture important aspects of the data generating process such as missingness, correlation, and over/underdispersion. This realization has motivated the development of a large class of estimating equations which account for these data characteristics and often yield improved inference for low-dimensional parameters. In this paper we introduce EEBoost, a novel strategy for variable selection and prediction which can be applied in any problem where inference would typically be based on estimating equations. The method is simple, flexible, and easily implemented using existing software. The EEBoost algorithm is obtained as a straightforward modification of the standard boosting (or functional gradient descent) technique. We show that EEBoost is closely related to a class of L_1 constrained projected likelihood ratio minimizations, and therefore produces similar variable selection paths to penalized methods without the need to apply constrained optimization algorithms. The flexibility of EEBoost is illustrated by applying it to simulated examples with correlated outcomes (based on generalized estimating equations) and time-to-event data with missing covariates (based on inverse probability weighted estimating equations). In both cases, EEBoost outperforms standard variable selection methods which do not account for the relevant data characteristics.

Keywords: Variable selection, model selection, prediction, estimating equations, boosting, projected likelihood

1 Introduction

In biomedical studies, it is common to obtain data where the number of predictors p greatly exceeds the number of individuals n for which these covariates are available. Many methods have been developed for performing variable/model selection in such $p > n$ problems. The vast majority of these methods are designed for uncomplicated regression setups such as simple (Efron et al., 2004; Tibshirani, 1996) and generalized (Park and Hastie, 2007) linear models, and proportional hazards/accelerated failure time models (Fan and Li, 2002). At the same time, a great deal of contemporary statistical literature has been dedicated to describing estimation methods which yield improved inference for low-dimensional parameters (i.e. $p < n$) within more complex frameworks, for example when there are missing outcomes or covariates, correlated observations, over/underdispersion, etc. Many of these methods are based on solving a set of estimating equations (Godambe, 1991).

In this paper, we seek to answer the question: Can the desirable properties of estimating equations be leveraged to improve the performance of variable/model selection procedures for complex data structures when the number of predictors is large in comparison to the sample size? We propose EE-Boost, a simple, general-purpose method for variable selection in any problem where low-dimensional estimation can be carried out via estimating equations

(EEs). We begin by introducing some notation, using it to describe the high-dimensional variable selection problem more formally. In Section 2, we describe the EEBoost algorithm, which is inspired by boosting algorithms from the machine learning literature. In Section 3, we motivate the use of EEBoost and discuss its relationship to existing methods by demonstrating its equivalence to the L_1-penalized minimization of a projected log likelihood ratio. Section 4 demonstrates the application of the EEBoost algorithm in two settings, via simulation: 1) Correlated outcomes (based on the Generalized Estimating Equations), and 2) Time-to-event data with covariates missing at random (based on Inverse Probability Weighted estimating equations). We conclude with a brief discussion and some open questions.

1.1 Setup and notation

Suppose we observe data $X_i \equiv (Y_i, Z_i), i = 1, \ldots, n$, where Y_i is some outcome of interest and Z_i is a vector of p covariates. Consider the problem of predicting future observations $Y_{n+1}, Y_{n+2}, \ldots Y_{n+K}$ based on their respective covariate vectors $Z_{n+1}, Z_{n+2}, \ldots Z_{n+K}$ with $[(Y_{n+1}, Z_{n+1}), (Y_{n+2}, Z_{n+2}), \ldots (Y_{n+K}, Z_{n+K})]$ arising from the same distribution F as $[(Y_1, Z_1), \ldots, (Y_n, Z_n)]$. Throughout this paper, we focus on the regression problem where some functional $\mathcal{G}$ of F is parametrized by $\beta \in \Theta \subset \mathbb{R}^p$ with $p < \infty$. The ultimate goal is to produce a set of coefficients $\hat{\beta}$ which minimizes the risk $E_F[L(X, \beta)] \equiv R(\beta)$ for some non-negative loss function L. Since the joint distribution F governing X is unknown, R cannot be computed directly. In practice, we hope to create

procedures generating $\hat{\beta}$ with $R(\hat{\beta}) \approx \min_\beta R(\beta) \equiv R_0$. It is well-known that when p is large in relation to n, procedures which base prediction on a selected subset of elements of β often yield better prediction than standard regression procedures which estimate the entire β vector. Such variable selection procedures also have the added benefit of identifying a manageable number of factors which may be worthy of further scientific study.

1.2 Previous work

In the past decade, there has been an explosion in the amount of statistical literature addressing the variable selection problem. The vast majority of techniques proposed in this literature apply to standard regression setups, mainly generalized linear models and simple survival models. Recently, attention has begun to shift towards variable selection procedures which account for complex data features (correlation, missingness, etc.), features that previously motivated the development of estimating equations for inference on low-dimensional parameters.

In some cases, it may be natural to pre- (or co-)process the available data so that standard variable selection procedures can be applied. For example, Yang et al. (2005) proposed methods which combine imputation and variable selection for performing model selection when covariates are missing at random. Such approaches are intuitive and relatively straightforward to implement, but are not easily generalized to settings where the data cannot be augmented or modified so that standard variable selection methods are applicable. On an-

other track, Fu (2003) and Johnson et al. (2008) have proposed a penalized estimating equation approach for performing variable selection in a large class of semi-parametric models. The resulting procedures have certain desirable optimality properties, but the computational complexity of the algorithms involved (both in terms of computational burden and difficulty of implementation) may limit their applicability in problems where p is truly large (eg. $p \approx 1,000$, not uncommon in many applications).

Researchers wishing to perform variable selection with complex, high-dimensional data structures may therefore face an unappealing choice: invest time and effort to adapt an existing variable selection method to their specific situation, or use available software which does not take important problem features into account. Our approach, EEBoost, is a computationally simple technique which approximates the behavior of penalized methods (the relationship is made explicit in Section 3) and is applicable whenever standard inference can be performed by solving a set of estimating equations.

2 EEBoost

The motivation for our proposed method is a technique often referred to as *boosting* or *functional gradient descent* (Freund and Schapire, 1997; Friedman, 2001). Boosting is an iterative procedure for building an additive model $\mathcal{G}^{(T)} = \sum_{j=1}^{J} h_j \cdot \beta_j^{(T)}$ for the functional $\mathcal{G}$. $\mathbb{H} = \{h_j, j = 1, \ldots, J\}$ is a set of candidate predictors, and $\beta_j^{(T)}$ is the coefficient after T iterations. Though $\mathbb{H}$ may be quite general in some boosting applications, we take $\mathbb{H} = \{Z_{\cdot j}, j = 1, \ldots, p\}$,

the set of available covariates in the columns of the design matrix Z. For a given loss function L, the basic boosting algorithm iteratively updates $\beta^{(T)}$ by incrementing the elements corresponding to directions in which the magnitude of the gradient of L is largest:

Algorithm 1 (Generic Boosting Algorithm)

1. *Set* $\beta^{(0)} = 0$
2. *For* $t = 1 : T$,
   (a) *Identify* $j_t = \arg\max_j |\frac{\partial L(X,\beta)}{\partial \beta_j}|$
   (b) *Set* $\beta_{j_t}^{(t)} = \beta_{j_t}^{(t-1)} - \alpha_t \ \text{sign}(\frac{\partial L(X,\beta)}{\partial \beta_{j_t}})$

The output from Algorithm 1 is a one-dimensional path $B = \{\beta^{(0)}, \ldots, \beta^{(T)}\}$ through Θ. Variable selection is achieved by "early stopping": Once B has been generated, one can apply a function $C : (X, B) \rightarrow \beta \in \Theta$ to choose a point on the path. If j_t is unique for all t (which generally occurs in practice), only one element of β is updated at each iteration, and $\beta^{(K)}$ can have at most K non-zero components. Hence, if C selects $\beta^{(M)}, M < p$, then at least $p - M$ variables have never been updated and hence are zero (i.e. "not selected"). The choice of C is discussed further in the following section.

In step 2b of Algorithm 1, α_t is a step length which gives the update increment in iteration t. Choosing $\alpha_t = \epsilon$, a small constant, has been shown to yield good results in practice (Bühlmann and Yu, 2003; Friedman, 2001).

2.1 EEBoost algorithm

If $X \sim F_\beta$, a known parametric distribution, a standard approach for estimating a low-dimensional β is to minimize the negative log-likelihood ℓ. For high-dimensional β, we may generate a variable selection path via Algorithm 1, using the score function $\partial\ell/\partial\beta$ in step 2. This is precisely the idea behind LogitBoost (Friedman et al., 2000), a variable selection and classification technique for binary outcomes based on the binomial likelihood.

When the distribution of X is unknown or cannot be written in closed form, low-dimensional parameter estimates are often obtained by solving a set of estimating equations $g(X, \beta) = 0$. Estimators defined by the solutions of these equations may be preferred to those derived by directly maximizing an incorrect log likelihood; for example, they may be more efficient than competing estimators or remain unbiased if certain relationships are misspecified (Lipsitz et al., 1994; Stefanski and Boos, 2002).

Estimating equations may not correspond to the gradient of any closed-form loss function, but they are generally obtained as modifications of such gradients and can be expected to behave similarly. The EEBoost algorithm, then, consists of substituting the vector of estimating equations $g(X, \beta)$ for $\partial L(X, \beta)/\partial\beta$ in Algorithm 1, yielding:

Algorithm 2 *EEBoost Algorithm*

1. *Set* $\beta^{(0)} = 0$
2. *For* $t = 1 : T$,

(a) Identify $j_t = \arg\max_j |g_j(X, \beta)|$

(b) Set $\beta_{j_t}^{(t)} = \beta_{j_t}^{(t-1)} - \epsilon \ \text{sign}(g_{j_t}(X, \beta))$

We emphasize that the EEBoost algorithm is technique for generating variable selection *paths*; it must be paired with a point chooser C to yield a variable selection procedure, the performance of which will depend on the choice of C. We purposely do not specify a general-purpose point chooser for EEBoost paths in this work; we believe that the choice of C should be driven by the loss function of scientific interest, and need not necessarily account for the manner in which the variable selection paths were generated. As an example, if EEBoost were applied to data comprising correlated observations on a number of individuals, depending on the scientific goals one might consider point choosers which treated either a) each observation or b) each individual as a separate unit. For those seeking a single technique applicable to a variety of estimating equations, we note the work of Pan (2001), who proposed a bootstrap-smoothed cross validation estimate of the expected predictive bias.

In the following section, we motivate the use of EEBoost more formally by showing its close relationship to a particular class of L_1 penalized methods.

3 Properties of EEBoost

3.1 Boosting and L_1 penalization

For a given loss function L and fixed β_0, consider the problems

$$\min_{\beta} L(X, \beta) \text{ subj to } ||\beta - \beta_0||_1 < \epsilon \tag{1}$$

$$\min_{\beta} L(X, \beta) \text{ subj to } ||\beta - \beta_0||_1 < \epsilon, ||\beta||_1 > ||\beta_0||_1 \tag{2}$$

as $\epsilon \to 0$. The solution β^* of (2) satisfies

$$\beta_j^* \neq \beta_{0,j}^* \Rightarrow \frac{\partial L}{\partial \beta_j}(\beta_0) = \max_j \left| \frac{\partial L}{\partial \beta_j}(\beta_0) \right|$$

provided $sign(\beta_{0j}) = -sign\left(\frac{\partial L}{\partial \beta_j}(\beta_0)\right)$. In other words, β^* represents a coordinate descent step from β_0. Let $\beta^\dagger$ denote the solution of (1). Then, if $\beta^\dagger$ satisfies the component-wise monotonicity condition $|\beta_j^\dagger| > |\beta_{0,j}|$ for all j, then $\beta^\dagger = \beta^*$.

As λ varies, the solutions to (1) obtained by varying β_0 (and letting $\epsilon \to 0$) are exactly those of the L_1 penalized problem

$$\min_{\beta} L(X, \beta) \text{ subj to } ||\beta||_1 \leq \lambda$$

i.e. the LASSO path (Tibshirani, 1996). The solutions to (2) correspond to the path generated by applying Algorithm 1 with gradient $g = \frac{\partial L}{\partial \beta}$. Hence, for sufficiently small ϵ, we would expect the LASSO and boosting paths of (1) and

(2) to coincide, provided the component-wise monotonicity condition holds. In practice, this monotonicity condition seems to be satisfied in many applications, and Hastie et al. (2001) and Efron et al. (2004) have demonstrated the resulting remarkable congruence between L_1 constrained and ϵ-boosting paths in several examples.

The EEBoost algorithm we propose makes use of estimating equations which are not necessarily the derivative of a closed-form loss function, so the correspondence to L_1 penalization is less clear. In the following section, we show that EEBoost can also be interpreted in terms of L_1 penalization by showing the equivalence between EEBoost solution paths and a sequence of L_1-constrained projected likelihood ratio minimizations.

3.2 EEBoost and L_1 penalization

In what follows, we restrict attention to the case where the first two moments of $Y_1, \ldots, Y_n$ are defined by $\mu_i(\theta)$ and $\Sigma_i(\theta)$, which are functions of $\theta \in \Theta \equiv \mathbb{R}^p$. In this semiparametric setup, inference for θ is often undertaken via the quasi-score

$$g(\theta) = \sum_{i=1}^{n} \mu_i'(\theta)(\Sigma_i(\theta))^{-1}(Y_i - \mu_i(\theta)) \tag{3}$$

where $\mu' = \frac{\partial \mu}{\partial \theta}$. $g(\theta)$ can be viewed as a projection of the true score function onto a space spanned by $Y_1 - \mu_1(\theta), \ldots, Y_n - \mu_n(\theta)$. The true likelihood $L(\theta)$ is unknown, but we may construct an approximation to it by projecting the likelihood ratio $\Lambda(\theta_0, \theta) = \frac{L(\theta_0)}{L(\theta)}$ onto a linear space defined by functions of the

form

$$c(\theta_0) + \sum_{k, i_1 < \cdots < i_k} a_{i_1 \cdots i_k}(\theta_0)(Y_{i_1} - \mu_{i_1}) \cdots (Y_{i_k} - \mu_{i_k})$$

Projecting $\Lambda(\theta_0, \theta)$ onto this linear space yields the *projected artificial likelihood ratio*

$$\lambda(\theta_0, \theta) = \prod_{i=1}^{n} \left[1 + (\mu_i(\theta) - \mu_i(\theta_0))(\Sigma_i(\theta_0))^{-1}(Y_i - \mu_i(\theta_0))\right] \tag{4}$$

where θ is assumed to lie in a neighborhood of the parameter value $\theta_0 = \arg\min_{\theta} L(\theta)$. The theory of projected artificial likelihoods is treated in detail in McLeish and Small (1992) and Small and Wang (2003) (see pp. 197-240).

We consider $\ell = \log(\lambda)$ as a function of θ for fixed θ_0, writing

$$\ell(\theta) = \sum_{i=1}^{n} \log\left[1 + (\mu_i(\theta) - \mu_i(\theta_0))(\Sigma_i(\theta_0))^{-1}(Y_i - \mu_i(\theta_0))\right] \tag{5}$$

In the same way that we expect the quasi-score to behave similarly to the true score function, we anticipate that the projected artificial log likelihood ratio $\ell(\theta)$ should be a useful surrogate for the true log likelihood ratio. Hence, the paths generated by EEBoost should approximate those obtained by solving a sequence of L_1-constrained projected artificial log likelihood ratio minimization problems. After establishing some notation, we present a sequence of theorems formalizing this intuition; the proofs appear in the Supplementary Materials. The theorems and proofs presented build on the foundational work of Rosset et al. (2004).

Let $Y = \{Y_1, \ldots, Y_n\}$ be a vector of scalar outcomes, and $Z = \{Z_{1\cdot}, \ldots, Z_{n\cdot}\}$ be the $n \times p$ matrix of covariates consisting of the stacked covariate vectors

associated with the elements of Y; we denote row i and column j of Z by $Z_{i\cdot}$ and $Z_{\cdot j}$, respectively. Suppose that $E(Y_i \mid Z_{i\cdot}) = \mu_i(\beta) = \phi(Z_{i\cdot}\beta) \equiv \phi(\eta_i(\beta))$ for some link function ϕ and $\beta \in \mathbb{R}^p$. Let the quasi-score g be defined as in (3), and $\ell(\beta) = \log \lambda(\beta_0, \beta)$ be the projected artificial log-likelihood ratio corresponding to g for some β_0. Note that

$$\frac{\partial \ell(\beta)}{\partial \beta}\Big|_{\beta=\beta_0} = g(\beta_0) \tag{6}$$

i.e. the projected artificial log likelihood ratio is tangent to the quasi-score at β_0.

Now, let $\beta_L(s)$ be defined by

$$\beta_L(s) = \arg\min_{\beta} \ell(\beta) \;\; \text{subj to} \;\; \sum_j |\beta_j| \leq s \tag{7}$$

and let $\beta_B(T)$ be the coefficient vector after T iterations of EEBoost with descent directions defined by $g(\beta)$ and step length ϵ on the path generated by EE. The following theorem gives conditions under which the direction of change of β_L and β_B coincide:

Theorem 1 *Consider starting the EEBoost algorithm at some point $\beta_L(s)$. Suppose that:*

[Condition 1] For $s < s_0$, $\beta_L(s)$ and $\beta_B(T)$ are monotone in s and T, respectively, (i.e. for all j, $|\beta_L(s)|_j \leq |\beta_L(s')|_j$ for $s < s'$ and similarly for $\beta_B(T)$)

[Condition 2] For $s < s_0$, $||\beta_L(s)||_1 = s$

[Condition 3] For all β in a neighborhood of $\beta_L(s)$ and for all $i = 1, \ldots, n$,

$$(\beta - \beta_L(s))' \left(\frac{\partial^2 \eta_i}{\partial \beta^2}(\beta_L(s)) \right) (\beta - \beta_L(s)) = o(|\beta - \beta_L(s)|)$$

[Condition 4] For all β in a neighborhood of $\beta_L(s)$ and for all $i = 1, \ldots, n$,

$$\lim_{\substack{T \to \infty \\ \epsilon \to 0 \\ T \cdot \epsilon \to 0}} \frac{\eta(\beta_B(T)) - \eta(\beta_L(s))}{T \cdot \epsilon} = \lim_{\Delta s \to 0} \frac{\eta(\beta_L(s + \Delta s)) - \eta(\beta_L(s))}{\Delta s}$$

(i.e. the limit on both sides exists and is unique).

Then

$$\frac{\beta_B(T) - \beta_L(s)}{T \cdot \epsilon} \to \nabla \beta_L(s) \quad as\ \epsilon \to 0, T \to \infty, T \cdot \epsilon \to 0$$

Theorem 1 establishes the equivalence between the local behavior of β_L and β_B for an "idealized" boosting algorithm wherein $\epsilon \to 0, T \to \infty$, and $T \cdot \epsilon \to 0$. Condition 1 is the component-wise monotonicity condition described in the previous section which is difficult to verify analytically, but commonly holds in practice. Condition 2 requires that no unconstrained minimum of ℓ have L_1 norm smaller than s_0. Condition 3 is relatively mild; for example, it will hold for the standard GLM link functions when the entries of the covariate matrix are bounded. The key to Theorem 1, then, is Condition 4. The following result is helpful in identifying situations where Condition 4 will hold.

Theorem 2 *Define $\mathcal{A} = \{j : |g_j(\beta_L(s))| = \max_j |g_j(\beta_L(s))|\}$. Suppose that conditions 1-3 of Theorem 1 hold, and that in addition:*

- *For all s, $|\mathcal{A}| < n$ (i.e. the number of elements in $\mathcal{A}$ is smaller than n)*

- *For all β in a neighborhood of $\beta_L(s)$ and for all $k = 1, \ldots, p$,*

$$\left(\frac{\partial g_k}{\partial \beta}(\beta_L(s))\right)' (\beta - \beta_L(s)) = O(|\beta - \beta_L(s)|) \tag{8}$$

$$(\beta - \beta_L(s))' \frac{\partial^2 g_k}{\partial \beta^2}(\beta - \beta_L(s)) = o(|\beta - \beta_L(s)|) \tag{9}$$

$$[\eta(\beta) - \eta(\beta_L(s))]' \left(\frac{\partial^2 \ell}{\partial \eta^2}(\beta_L(s))\right) [\eta(\beta) - \eta(\beta_L(s))] = o(|\eta(\beta) - \eta(\beta_L(s))|) \tag{10}$$

- *η is continuous at $\beta_L(s)$*

Then a sufficient condition for Condition 4 of Theorem 1 to hold is that

$$sign(g_k(\beta_L(s)) \frac{\partial^2 \mu_i}{\partial \beta_j \partial \beta_k}(\beta_L(s)) \text{ does not depend on } k \tag{11}$$

for $j, k \in \mathcal{A}$ and $i = 1, \ldots, n$.

There are two important special cases where (11) holds:

1. *A linear model where $\mu(\beta) = Z\beta$.* In this case, $\frac{\partial^2 \mu_i}{\partial \beta_j \partial \beta_k} = 0$, and hence the condition holds trivially.

2. *$\mathcal{A}$ has only one element.* This will occur if the maximum element of $|g(\beta_L(s))|$ is unique. In practice, the entries of g are generally distinct, and therefore we observe concordance between the sequence of solutions

to the L_1 penalized problems and the path generated by boosting with small step length.

Given the widespread success of L_1-penalized methods for variable selection and prediction, it is encouraging that EEBoost approximates the solution path for a particular sequence of L_1 penalized problems. In the next section, we demonstrate the flexibility of EEBoost by applying it to perform variable selection and prediction in realistic scenarios.

4 Example Applications

4.1 Correlated outcomes

To test the performance of the EEBoost algorithm on correlated outcome data, we simulated data from $n = 30$ individuals, with four observations per individual. For each observation $i = 1, \ldots, 30$, $j = 1, \ldots, 4$, a vector of covariates Z_{ij} of length 100 was simulated according to a multivariate normal distribution with mean zero, and covariance matrix Σ_Z defined by $Var(Z_{ijk}) = 0.25, Corr(Z_{ijk}, Z_{ijl}) = 0.3, k \neq l$ (before running the algorithms, each column of the covariate matrix was standardized to have mean zero and unit variance). Outcomes Y_{ij} were generated from a multivariate normal distribution with mean $\mu_i = Z_i'\beta$, and an exchangeable correlation structure defined by $Var(Y_{ij}) = 1, Corr(Y_{ij}, Y_{ik}) = \rho, j \neq k$. The entries $(\beta_1, \beta_2, \ldots, \beta_{100})$ of the

coefficient vector β were set as

$$\beta_m = 0.5,\ 1 \leq m \leq 5$$

$$\beta_m = 0.2,\ 6 \leq m \leq 10$$

$$\beta_m = 0.05,\ 11 \leq m \leq 20$$

$$\beta_m = 0,\ 21 \leq m \leq 100$$

We applied the following four algorithms to the generated data:

1. **EE(Ind):** EEBoost based on the Generalized Estimating Equations (GEE) (Liang and Zeger, 1986) with independence working correlation matrix.

2. **EE(Exch):** EEBoost based on the GEEs with exchangeable working correlation matrix, correlation parameter assumed known.

3. **EE(Est):** EEBoost based on the GEEs with exchangeable working correlation matrix, correlation parameter estimated at each iteration using the current coefficient vector.

4. **LARS:** Least Angle Regression (Efron et al., 2004), a fast implementation of the LASSO for linear models. Does not account for correlation of observations.

For each simulation, the four algorithms produced variable selection paths $\hat{\beta}$. In a small number of simulation runs, the EE(Est) procedure encountered

numerical problems and oscillated between two parameter values; the tables and plots which follow exclude these runs.

4.1.1 Prediction

We estimated prediction error at points on these paths by generating 100 test datasets with the same settings as above and averaging the residual sum of squares over these test sets. Figure 1 displays the estimated prediction error as a function of $||\hat{\beta}||_1$ for the four methods when $\rho = 0, 0.3, 0.7$, and 0.9. Note that Figure 1 provides estimates of the prediction error for a large number of points on the paths generated by the four algorithms; if the prediction error curve for one algorithm lies below that of another, this suggests that when paired with a suitable point chooser, the former algorithm will have smaller prediction error than the latter.

Table 1 summarizes the simulation results in a different way: In each simulation, the estimated prediction error is calculated at each point on the variable selection paths, the minimum prediction error on the path is computed, and the minima for each algorithm are ranked. Table 1 presents the mean and standard deviation of these ranks over the simulations. The results give the expected relative performance of the various algorithms when paired with "ideal" point choosers, i.e. those for which $R(C(X, B)) = \min_{\beta \in B} R(\beta)$.

Table 1 and Figure 1 illustrate that prediction error is lower when variable selection takes into account the correlation structure of the outcomes, and the improvement achieved increases with the correlation. When $\rho = 0.9$,

Figure 1: Prediction error as a function of $||\hat{\beta}||_1$ for four variable selection algorithms. Short dashed line = EE(Ind), solid = EE(Exch), dotted = EE(Est), long dashed = LARS

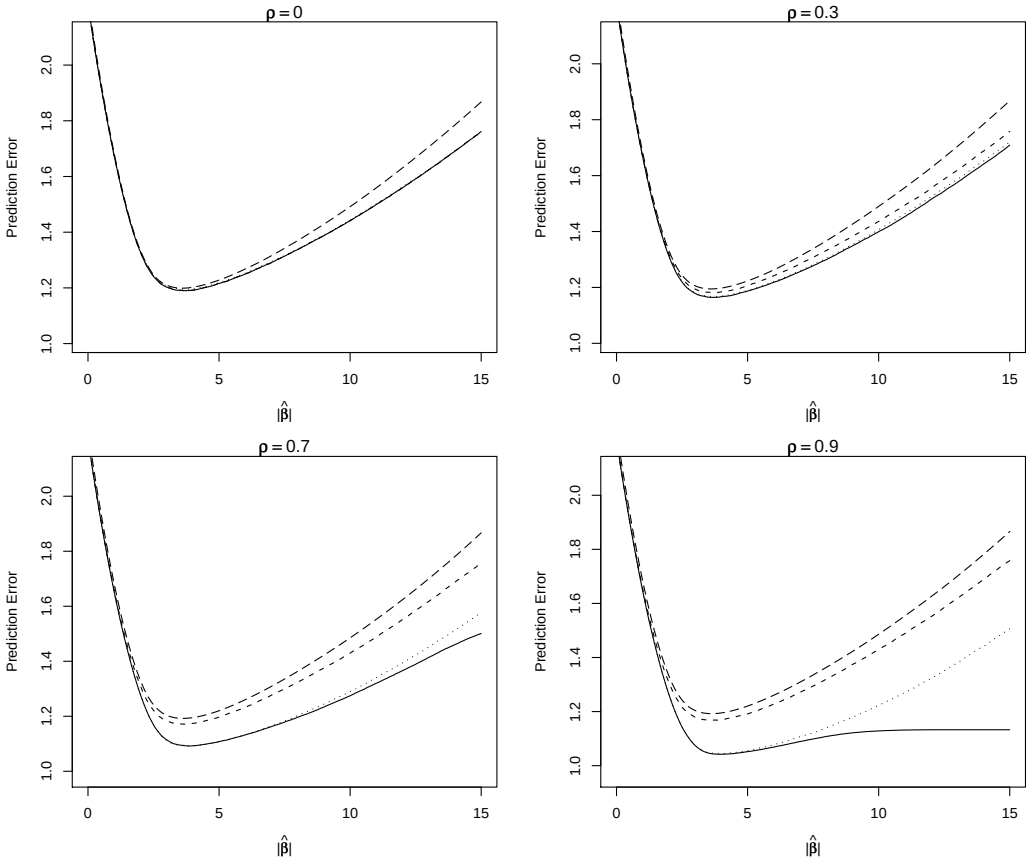


Table 1: Mean rank (SD) of minimum prediction error for four variable selection methods. ρ = correlation between intra-individual observations.

ρ	EE(Ind)	EE(Exch)	EE(Est)	LARS
0	2.02 (0.55)	2.02 (0.55)	2.29 (1.18)	3.68 (0.71)
0.3	2.57 (0.83)	1.76 (0.88)	1.93 (0.83)	3.74 (0.65)
0.7	2.81 (0.52)	1.38 (0.54)	2.04 (1.02)	3.77 (0.46)
0.9	2.66 (0.54)	1.21 (0.4)	2.57 (1.2)	3.56 (0.54)

for example, Figure 1 shows that the prediction error achieved at $||\hat{\beta}||_1 = 4$ (near the minima of the plotted prediction error curves) is approximately 15% lower for EE(Exch) and EE(Est) than for EE(Ind) and LARS. Though the curves for EE(Exch) and EE(Est) appear to coincide, Table 1 reveals that EE(Exch) performs more consistently, as it generally achieves a smaller minimum prediction error than the other three methods. EE(Est) achieves its best performance when correlation between outcomes is moderate. Allowing for more generality in the correlation structure may degrade the performance of EE(Est) relative to EE(Exch) and other approaches.

4.1.2 Variable selection

Figure 2 summarizes the variable selection performance of EE(Ind), EE(Exch), EE(Est), and LARS. For a given point $\hat{\beta}_k$ on the variable selection path, sensitivity *sens* and specificity *spec* were computed via

$$sens = \frac{\sum_{j=1}^{p} \mathbb{1}_{\left[\hat{\beta}_{kj} \neq 0\right]}}{\sum_{j=1}^{p} \mathbb{1}_{[\beta_{0j} \neq 0]}}$$

$$spec = \frac{\sum_{j=1}^{p} \mathbb{1}_{\left[\hat{\beta}_{kj} = 0\right]}}{\sum_{j=1}^{p} \mathbb{1}_{[\beta_{0j} = 0]}}$$

where β_0 is the length-p parameter vector used to generate the data. Figure 2 plots the empirical means (over 100 simulations) of *sens* and *spec* versus $||\hat{\beta}||_1$.

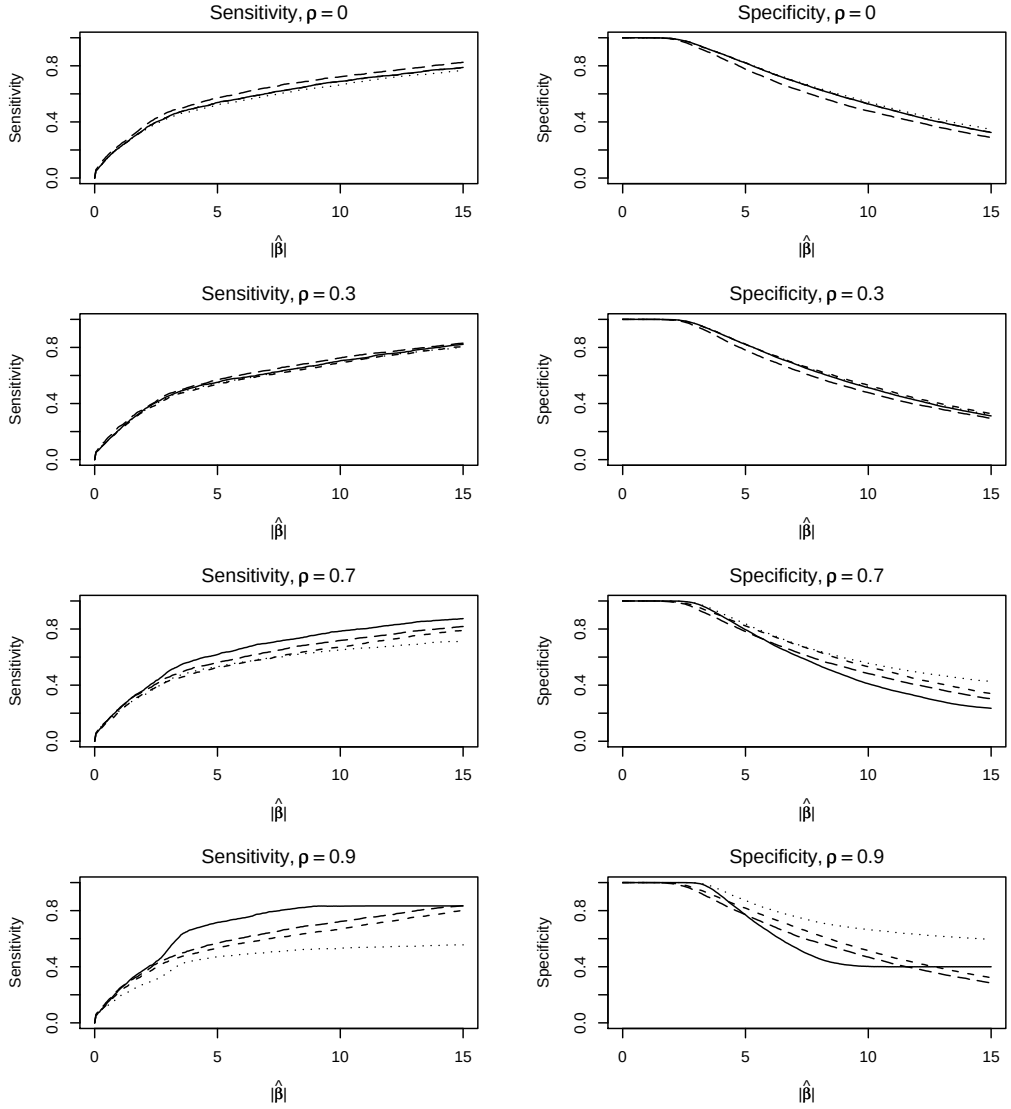
We observe that EE(Exch) has slightly higher sensitivity than competing methods for larger values of ρ. Specificity is comparable for all methods except when $\rho = 0.9$, where the specificity of EE(Exch) decreases more rapidly with $||\hat{\beta}||_1$ than other methods. The flat sensitivity and specificity trajectories of EE(Est) are likely related to the numerical problems described above.

4.2 Time-to-event data with missing covariates

Next, we apply the EEBoost algorithm to the problem of variable selection when the outcomes are survival/censoring times and some of the covariates are missing. There are separate literatures on fitting low-dimensional proportional hazards models when covariates are missing (Lin and Ying, 1993; Paik and Tsai, 1997; Wang and Chen, 2001; Chen, 2002) and on the problem of variable selection when covariates are completely observed (Fan and Li, 2002; Tibshirani, 1997), but we are not aware of any techniques for tackling the problem we consider here.

Let T^0 and C^0 be survival and censoring times, let $T = \min(T^0, C^0)$, and define $\delta = \mathbb{1}_{[T^0 \leq C^0]}$. We assume that each subject i has a vector of covariates (Z_i, X_i), where the Z_i are always observed and the X_i are either completely observed ($R_i = 1$) or completely unobserved - such a scenario might arise, for example, when the X_i represent expensive biological measurements. We

Figure 2: Variable selection sensitivity and specificity for four variable selection algorithms. Short dashed line = EE(Ind), solid = EE(Exch), dotted = EE(Est), long dashed = LARS



assume further that T^0 and C^0 are independent given (Z, X), and that the covariate values are missing at random ($R \perp\!\!\!\perp X \mid (T, \delta, Z)$).

The Cox proportional hazards model (Cox, 1972) is commonly employed to analyze survival data, with inference based on the partial likelihood score equations. When some covariate values are missing, but the probabilities of missingness $\pi_i = P(R_i = 1 \mid T_i, \delta_i, Z_i)$ are known, valid inference may be obtained by solving the inverse probability weighted estimating equations $g_{IPW}(\beta) = 0$ (see Wang and Chen (2001)). The EEBoost iteration we employ updates β according to which element of the vector $g_{IPW}(\beta)$ is largest in magnitude.

We assume a setup with $n = 150$ subjects on which $p = 40$ covariates are (possibly) observed. Let W be an $n \times p$ covariate matrix with rows generated as independent multivariate $N(0, 0.5I_{p\times p})$ vectors. Failure times were generated from an exponential distribution with rate $\eta = \theta^0 + W'\beta$, with $\theta^0 = -5$ and

$$\beta_1 = \beta_2 = 1, \beta_3 = \beta_4 = 0.5, \beta_5 = 0.25, \beta_6 = \cdots = \beta_{10} = 0.15$$
$$\beta_{11+k} = \beta_k, \ k = 0, \ldots, 10$$
$$\beta_{21} = \cdots = \beta_{40} = 0$$

Censoring times were generated to give a censoring rate of approximately 33%. The covariates were split into two sets, X and Z, according to whether or not they were subject to missingess; we set X to be the first covariates 31-40 (i.e. the last ten columns of W), while the remaining ("always observed") covariates comprised Z. For each individual i, X_i was assumed to be missing according

to

$$\pi_i \equiv P(R_i = 1 \mid Z_i, T_i, \delta_i) = \text{expit}(\alpha + \mathbb{1}_{[T_i > m_T]}\gamma' Z_i + \mathbb{1}_{[T_i \leq m_T]}\zeta' Z_i)$$

where m_T is the median of the observed survival/censoring times, and

$$\gamma_1 = \cdots = \gamma_5 = 1.5, \gamma_6 = \cdots = \gamma_{40} = 0$$
$$\zeta_1 = \cdots = \zeta_5 = 0, \zeta_6 = \cdots = \zeta_{10} = -1.5, \zeta_{11} = \cdots = \zeta_{40} = 0$$

The overall rate of missingness was controlled by the intercept term α. We considered scenarios with $\alpha = -6, -4, -2$ and 0, yielding average missingness probabilities of approximately 8%, 21%, 41%, and 66% , respectively. The EEBoost algorithm was run for 500 iterations with a stepwise increment of $\epsilon = 0.03$.

We compared three versions of EEBoost with two alternative approaches:

- EE(Fixed): The described EEBoost algorithm with π_i values assumed fixed and known.
- EE(Est_1): The described EEBoost algorithm with π_i values estimated from a logistic model with Z, $\mathbb{1}_{[T > m_T]}$, and their interaction included as linear predictors (i.e. the model for calculating probability of missingness is specified correctly).
- EE(Est_2): The described EEBoost algorithm with π_i values estimated from a logistic model with $\log(T+1)$ included as the only linear predictor (i.e. the model for calculating probability of missingness is specified

incorrectly).

- CoxPath(Full): A method based on minimizing the L_1-penalized Cox partial likelihood (fitted using method `coxpath` from R package `glmpath`). This version of the algorithm uses the full covariate matrix, without any missing values.

- CoxPath(CC): Same as previous, but only the subjects with complete covariate data are analyzed.

4.2.1 Prediction

Figure 3 and Table 2 summarize the performance of the five algorithms in the same manner described above, but using a different metric to assess predictive accuracy. Let $\hat{\beta}_M^{(k)}$ be a parameter vector estimate with L_1 norm k derived by method M ($M \in \{$ EE(Fixed), EE(Est$_1$), ... ,CoxPath(CC) $\}$). Assuming (correctly) that the baseline hazard is exponential, the predicted median failure time for individual i is

$$\hat{m}_i^{(k)}(M) = \frac{\log(2)}{\exp(\theta^0 + W_i' \hat{\beta}_M^{(k)})}$$

where $W_i = (X_i, Z_i)$ is the full covariate vector for individual i without any missing values. Let T_i^0 be that individual's true failure time. Let $\hat{R}_M^{(k)} = rank(\hat{m}_1^{(k)}(M), \ldots, \hat{m}_n^{(k)}(M))$ and $R^0 = rank(T_1^0, \ldots, T_n^0)$ be the vector of ranks of the predicted median (at boosting iteration k) and true survival times.

Table 2: Mean rank (SD) of $\max_k C^{(k)}_{rnk}$ for five variable selection methods. $P_{R=0}$ = proportion of individuals with missing X.

α	$P_{R=0}$	EE(Fixed)	EE(Est_1)	EE(Est_2)	CoxPath(Full)	CoxPath(CC)
-6	0.08	3.96 (0.92)	2.56 (0.96)	2.17 (0.86)	4.51 (1.05)	1.81 (0.94)
-4	0.21	3.53 (0.93)	2.93 (1.08)	1.85 (0.72)	4.85 (0.48)	1.84 (0.94)
-2	0.41	3.5 (0.78)	2.92 (0.93)	1.62 (0.79)	5 (0)	1.96 (0.83)
0	0.66	3.29 (0.88)	2.81 (1.00)	1.76 (0.91)	5 (0)	2.13 (0.98)

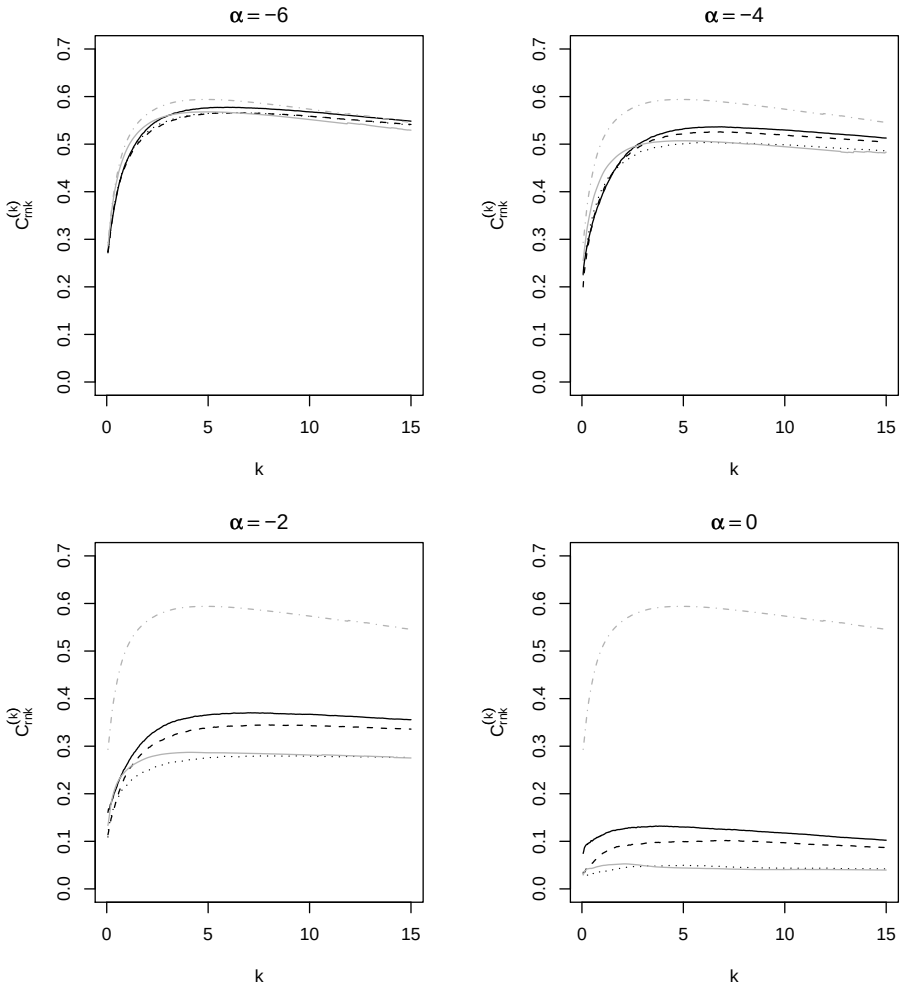
Then we define

$$C^{(k)}_{rnk}(M) = corr(\hat{R}^{(k)}_M, R^0)$$

Figure 3 plots $C^{(k)}_{rnk}$ versus k for each of the five methods, and Table 2 reports the mean and standard deviation of the ranks of $\max_k C^{(k)}_{rnk}$, over 100 simulations. Note that larger ranks correspond to larger maxima, indicating a method which produces predicted survival times agreeing more closely with the true survival times.

The results from Table 2 and Figure 3 show that versions of EEBoost where the missingness probabilities are fixed (EE(Fixed)) or estimated from a correct model (EE(Est_1)) outperform the method based only on the complete cases (CoxPath(CC)). When the missingness probabilities are relatively low, EE(Fixed) and EE(Est_1) perform nearly as well as a procedure which uses the full data (CoxPath(Full)), but as expected their performance relative to this hypothetical gold standard degrades as the proportion of observations with missing covariate values increases. When the model governing the probability of missingness is misspecified (EE(Est_2)), EEBoost offers no performance gain over a complete case analysis.

Figure 3: $C_{rnk}^{(k)}$ as a function of k for five variable selection algorithms. Gray lines represent CoxPath(Full) (dot dashed) and CoxPath(CC) (solid). Black lines represent EE(Fixed) (solid), EE(Est_1) (dashed), and EE(Est_2) (dotted).



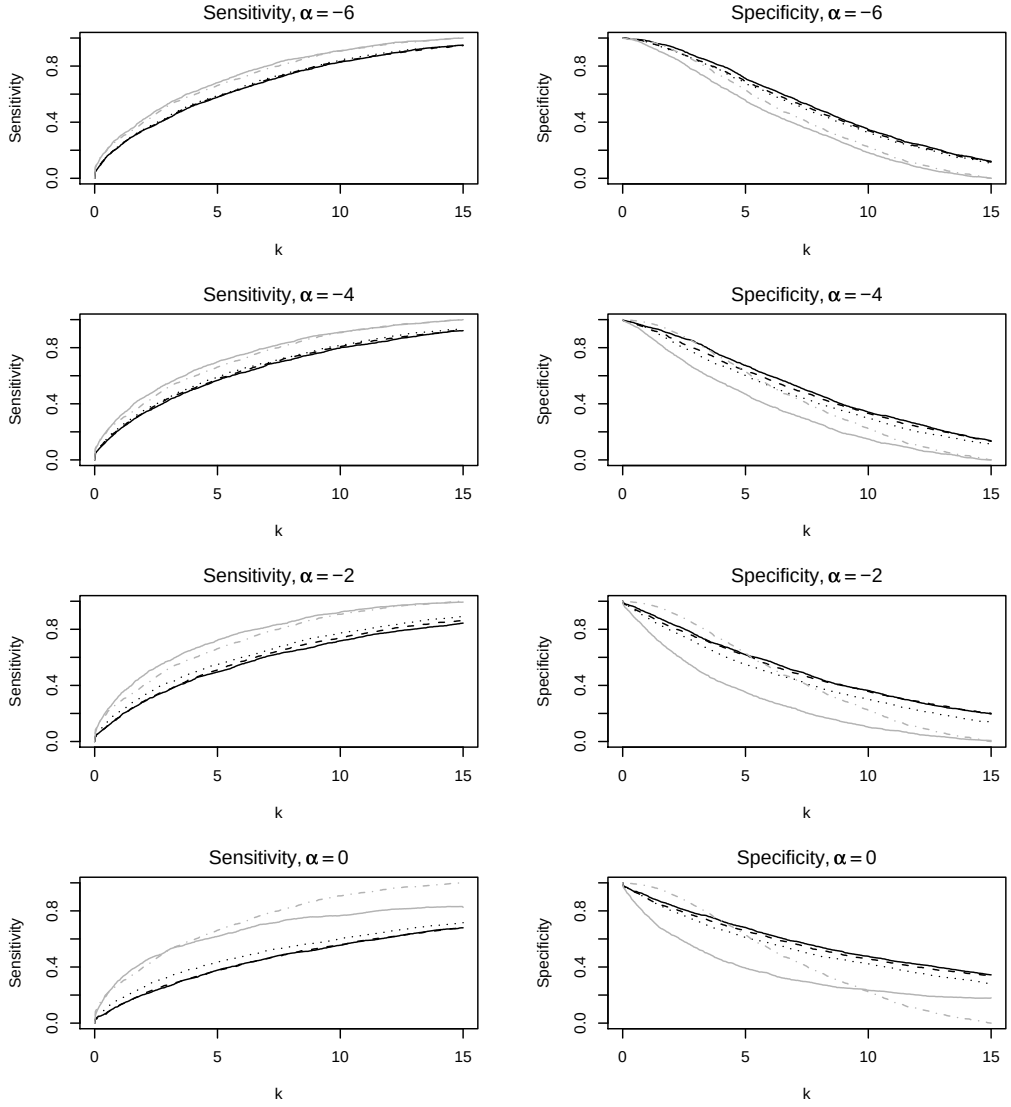
4.2.2 Variable Selection

Figure 4 presents the variable selection performance of the five algorithms using the measures of sensitivity and specificity described in the previous section. The CoxPath(Full) and CoxPath(CC) paths generally have higher sensitivity and lower specificity than the EEBoost paths, suggesting that the latter are generating paths with more sparsity (i.e. fewer nonzero coefficients). The three EEBoost paths have very similar sensitivity and specificity across the range of k. Though the CoxPath(Full) path (gray dot-dashed line) has lower variable selection specificity than the EEBoost paths for $k > 5$, it has higher or comparable specificity for smaller values of k where $C_{rnk}^{(k)}$ is largest.

5 Discussion

One of the main features of the EEBoost algorithm is its simplicity. Beginning with available code for solving a set of estimating equations (say for the purposes of low-dimensional estimation), a variable selection path can be generated with minimal effort. Investigators are free to apply any appropriate point choosing procedure (eg. cross and holdout set validation, various information criteria) to the resulting path. EEBoost is also flexible, allowing modifications which can change the behavior of the algorithm in order to accommodate problem-specific restrictions and features. For example, it is trivial to modify EEBoost so that constraints on coefficient values are obeyed, or so that certain variables are always included in the model. One could also consider adapting

Figure 4: Variable selection sensitivity and specificity for five variable selection algorithms. Gray lines represent CoxPath(Full) (dot dashed) and CoxPath(CC) (solid). Black lines represent EE(Fixed) (solid), EE(Est_1) (dashed), and EE(Est_2) (dotted).



ideas suggested by Friedman and Popescu (2004) to yield paths indexed by an additional parameter controlling the number of coefficients which are updated at each iteration.

Though we have given theoretical results and illustrated the application of EEBoost in the context of estimating equations which are quasi-scores, its use is not restricted to these setups. In other work, we have successfully applied EEBoost with the augmented inverse probability weighted (AIPW) estimating equations of Rotnitzky and Robins (1995). Generally, we would expect EEBoost to perform well whenever the relative magnitudes of the elements of the estimating equation vector reflect the explanatory ability of the variables to which they correspond. Further work is needed to characterize how the structure of the data, the chosen loss function, and the operating characteristics of the estimating equations influence the performance of EEBoost, and identify the contexts in which EEBoost will provide the most benefit over competing methods.

Acknowledgment

Many thanks to Peter Gilbert, who provided both guidance and invaluable feedback while this work was being developed.

References

Bühlmann, P. and Yu, B. (2003). Boosting with the l loss: Regression and classification. *Journal of the American Statistical Association* **98,** 324–339.

Chen, H. Y. (2002). Double-semiparametric method for missing covariates in cox regression models. *Journal of the American Statistical Association* **97,** 565–576.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)* **34,** 187–220.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics* **32,** 407–451.

Fan, J. and Li, R. (2002). Variable selection for cox's proportional hazards model and frailty. *The Annals of Statistics* **30,** 74–99.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **55,** 119–139.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *The Annals of Statistics* **28,** 337–374.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* **29,** 1189–1232.

Friedman, J. H. and Popescu, B. E. (2004). Gradient directed regularization for linear regression and classification. Technical report, Stanford University.

Fu, W. J. (2003). Penalized estimating equations. *Biometrics* **59,** 126–132.

Godambe, V. P., editor (1991). *Estimating Functions (Oxford Statistical Science Series)*. Oxford University Press, USA.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer.

Johnson, B. A., Lin, D. Y., and Zeng, D. (2008). Penalized estimating functions and variable selection in semiparametric regression models. *Journal of the American Statistical Association* **103,** 672–680.

Liang, K.-Y. and Zeger, S. L. (1986). Longitudinal data analysis using generalized linear models. *Biometrika* **73,** 13–22.

Lin, D. Y. and Ying, Z. (1993). Cox regression with incomplete covariate measurements. *Journal of the American Statistical Association* **88,** 1341–1349.

Lipsitz, S. R., Fitzmaurice, G. M., Orav, E. J., and Laird, N. M. (1994). Performance of generalized estimating equations in practical situations. *Biometrics* **50,** 270–278.

McLeish, D. L. and Small, C. G. (1992). A projected likelihood function for semiparametric models. *Biometrika* **79,** 93–102.

Paik, M. C. and Tsai, W. Y. (1997). On using the cox proportional hazards model with missing covariates. *Biometrika* **84,** 579–593.

Pan, W. (2001). Model selection in estimating equations. *Biometrics* **57,** 529–534.

Park, M. Y. and Hastie, T. (2007). L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **69,** 659–677.

Rosset, S., Zhu, J., and Hastie, T. (2004). Boosting as a regularized path to a maximum margin classifier. *Journal of Machine Learning Research* **5,** 941–973.

Rotnitzky, A. and Robins, J. M. (1995). Semi-parametric estimation of models for means and covariances in the presence of missing data. *Scandinavian Journal of Statistics* **22,** 323–333.

Small, C. G. and Wang, J. (2003). *Numerical Methods for Nonlinear Estimating Equations*. Clarendon Press - Oxford.

Stefanski, L. A. and Boos, D. D. (2002). The calculus of m-estimation. *The American Statistician* **56,** 29–38.

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

Tibshirani, R. (1997). The lasso method for variable selection in the cox model. *Statistics in Medicine* **16,** 385–395.

Wang, C. Y. and Chen, H. Y. (2001). Augmented inverse probability weighted estimator for cox missing covariate regression. *Biometrics* **57,** 414–419.

Yang, X., Belin, T. R., and Boscardin, W. J. (2005). Imputation and variable selection in linear regression models with missing covariates. *Biometrics* **61,** 498–506.